UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Denise Bendo Demétrio

Infra-Estrutura de Protocolação Digital de Documentos Eletrônicos

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação.

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis, Dezembro de 2003

Infra-Estrutura de Protocolação Digital de Documentos Eletrônicos

Denise Bendo Demétrio

Esta Dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Raul Sidnei Wazlawick, Dr.

Coordenador do Curso

Banca Examinadora

_____

Prof. Ricardo Felipe Custódio, Dr.

Orientador

_____

Prof. Daniel Santana de Freitas, Dr.

_____

Prof. Carlos Roberto de Rolt, Dr.

_____

Prof. Jeroen Antonius Maria van de Graaf, Dr.

_____

Prof. Ricardo Dahab, Dr.

“ Senhor, mostra-me o teu caminho e eu me conduzirei
segundo a tua verdade.
Unifica o meu coração para que ele tema o teu nome.
Senhor meu Deus, quero celebrar-te de todo o meu
coração, e glorificar o teu nome para sempre. ”
Salmo 86, 11-12

Aos meus pais, Valmir e Maria do Carmo.

Agradecimentos

"*Agradecer... Devolver a Deus o que veio dEle através das mãos dos irmãos.*"

- Ziza Fernandes

A Deus, por todas as bençãos que tenho recebido ao longo de minha vida.

A meus pais, Valmir e Maria do Carmo, pelo amor, carinho e apoio que sempre me deram e aos meus irmãos, Kétner e Vagner, pela compreensão e força.

A meu namorado, Cleyzer, pelo amor e companheirismo.

A todos os meus amigos que de uma forma ou outra contribuíram para esta conquista, em especial minhas amigas Débora Cabral Nazário, Vanessa Costa e Andréa Rosada. Não podendo deixar de citar minha amiga Camile Pimpão, por sua ajuda para a realização da defesa deste trabalho.

Não posso deixar de agradecer a secretária da pós graduação Vera Lucia Sodré pela imensa simpatia e prazer em ajudar as pessoas. Obrigada pelos “galhos quebrados”.

Quero agradecer ao Prof. Ricardo Felipe Custódio, por muito ter me ajudado a superar os obstáculos encontrados ao longo da realização deste trabalho. Agradeço também aos professores da banca de Trabalho Individual (TI) e aos professores da banca de dissertação por me ajudarem a melhorar a qualidade do trabalho.

Aos colegas do LabSEC, principalmente a Júlio da Silva Dias. À empresa Bry Tecnologia de Florianópolis pelas dúvidas sanadas.

Agradeço aos alunos de graduação Bruno Leonardo Martins de Melo e Vitor Claudino dos Santos por ajudarem a realizar a parte prática do trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

AC	Autoridade Certificadora.
ACTS	Serviço Automatizado de Tempo em Computador (*Automated Computer Time Service*).
AD	Autoridade de Datação.
ASCII	Código Americano Padrão para Troca de Informação (*American Standard Code for Information Interchange*).
ASN.1	Notação de Sintaxe Abstrata Um (*Abstract Syntax Notation One*).
AR	Autoridade de Registro.
BCD	Código Binário Decimal (*Binary Coded Decimal*).
BGIP	Base Gerencial de Informações de Protocolação.
DES	Padrão de Cifragem de Dados (*Data Encryption Standard*).
DH	Protocolo de troca de chaves Diffie-Hellman.
GMT	Tempo Médio de Greenwich (*Greenwich Mean Time*).
GPS	Serviço de Posicionamento Global (*Global Positioning Service*).
ICP	Infra-estrutura de Chaves Públicas.
IP	Protocolo da Internet (*Internet Protocol*).
IPv4	IP versão 4 (possuindo um campo de endereçamento de 32 bits).
IPv6	IP versão 6 (possuindo um campo de endereçamento de 128 bits).
IRIG	Grupo de Instrumentação Inter-Abrangência (*Inter-Range Instrumentation Group*).
LAN	Rede de Área Local (*Local Area Network*).
LabSEC	Laboratório de Segurança em Computação - UFSC.

LCR	Lista de Certificados Revogados.
MAC	Código de Autenticação de Mensagens (*Message Authentication Code*).
MD5	Algoritmo de Compilação de Mensagens (*Message-Digest Algorithm*).
MIB	Base de Informações Gerenciais (*Management Information Base*).
MIT	Instituto de Tecnologia de Massachusetts (*Massachusetts Institute of Technology*).
NAK	Reconhecimento Negativo (*Negative Acknowledgement*).
NIST	Instituto Nacional de Padrões e Tecnologia (*National Institute of Standards and Technology*).
NTP	Protocolo de Tempo em Rede (*Network Time Protocol*).
ON	Observatório Nacional.
PDDE	Protocoladora Digital de Documentos Eletrônicos.
PKCS	Padrões para Criptografia de Chave Pública (*Public Key Cryptography Standards*).
PSGPD	Protocolo Simples de Gerenciamento de Protocolação Digital.
RFC	Requisição para Comentários (*Request For Comments*).
RSA	Rivest-Shamir-Adleman.
SBS	Segundo Binário Exato (*Straight Binary Seconds*).
SNMP	Protocolo Simples de Gerenciamento de Rede (*Simple Network Management Protocol*).
SNTP	Protocolo de Tempo em Rede Simplificado (*Simple Network Time Protocol*).
SSL	Camada de Conexão Segura (*Secure Sockets Layer*).
TCP	Protocolo de Controle de Transmissão (*Transmission Control Protocol*).
TSP	Protocolo de Carimbo de Tempo (*Time-Stamp Protocol*).
UDP	Protocolo de Pacote do Usuário (*User Datagram Protocol*).
UTC	Tempo Universal Coordenado (*Universal Time Coordinated*).
WAN	Rede de Grande Área (*Wide Area Network*).

Lista de Símbolos

Auth.K_{id}	Campo de autenticação para a chave K_{id}.
Auth.MAC	Campo de autenticação.
C_0	Ponto de confiança inicial.
H_n	Resumo de um documento de ordem “n”.
H(X_n*)*	Resumo do documento X_n.
$IP_{cliente}$	Endereço IP do cliente.
$IP_{servidor}$	Endereço IP do servidor.
K_{id}	Chave de “id” para troca de mensagens.
K_{sessao}	Chave de sessão.
L_n	Encadeamento de ordem “n”.
$N_{randomico}$	Número randômico.
P_i	Participante de ordem “i”.
R_r	Rodada “r”.
S_0	Ponto de sincronismo inicial.
sig_A	Assinatura Digital feita por “A”.
t_n	Data e hora corrente.
X_n	Documento de ordem “n” a ser datado.
y_r	Resumo de ordem “r”.
Z	Ponto de cruzamento.

Resumo

A Autoridade de Datação (AD), responsável por protocolar documentos eletrônicos, possui capacidade limitada em relação ao número de protocolações realizadas em um mesmo intervalo de tempo. Isto inviabiliza a protocolação digital em organizações que possuem grande demanda de requisições. Este trabalho propõe uma infra-estrutura de protocolação digital que visa suprir esta necessidade através de melhorias feitas no protocolo de sincronismo de relógio *Network Time Protocol* (NTP) em relação à sua segurança e através de protocolação cruzada entre ADs. Uma infra-estrutura de protocolação digital permite distribuir as requisições, evitando a sobrecarga de uma única AD, bem como comparar temporalmente dois documentos protocolados em diferentes ADs.

Abstract

A Time-Stamping Authority (TSA), responsible for time-stamping electronic documents, has a limited capability in relation to the number of time-stamps carried out within a certain time interval. This limitation invalidates time-stamping in organizations that have a great demand in terms of requests. This study proposes a digital time-stamping infrastructure for electronic documents that intends to attend to this necessity through improvements to the Network Time Protocol (NTP) in relation to its security and by crossed time-stamping between two TSAs. A digital time-stamping infrastructure permits the distribution of time-stamping so avoiding the overworking of one TSA, as well as the temporal comparison of two documents time-stamped by different TSAs.

Capítulo 1

Introdução

Há muitas vantagens do documento eletrônico em relação aos tradicionais documentos em papel. Os documentos eletrônicos são normalmente fáceis de gerenciar, ocupam menos espaço físico e podem ser transmitidos de forma rápida através das redes de comunicação de dados.

Contudo, foi necessário o desenvolvimento de tecnologias que possibilitassem agregar ao documento eletrônico a mesma confiança que já existe em relação ao documento em papel. Isso é possível a partir do uso de mecanismos criptográficos que possibilitam a autenticação, a integridade e a tempestividade[1] dos documentos eletrônicos, criando-se o chamado documento eletrônico seguro. A existência destes documentos tem promovido a virtualização de inúmeras aplicações, anteriormente só possíveis usando meios materiais de suporte à informação.

Para agregar ao documento eletrônico a mesma segurança que os documentos tradicionais possuem, três elementos devem ser considerados confiáveis e seguros, conforme ilustra a Figura 1.1. Estes três elementos são:

- O certificado da Autoridade Certificadora[2](AC) deve ser confiável e válido pois este certificado é a raiz de todos os outros certificados emitidos pela AC. Um certificado digital utilizado para assinar um documento eletrônico seguro garante a autenticação e integridade do mesmo (STALLINGS, 1998);

[1]Tempestividade: Relacionado ao tempo (AURÉLIO, 1999).

[2]Autoridade Certificadora: Autoridade que emite e assina certificados digitais.

- O documento deve possuir a data e hora confiáveis do momento que foi criado ou assinado;

- Os algoritmos de criptografia não podem ser quebrados.

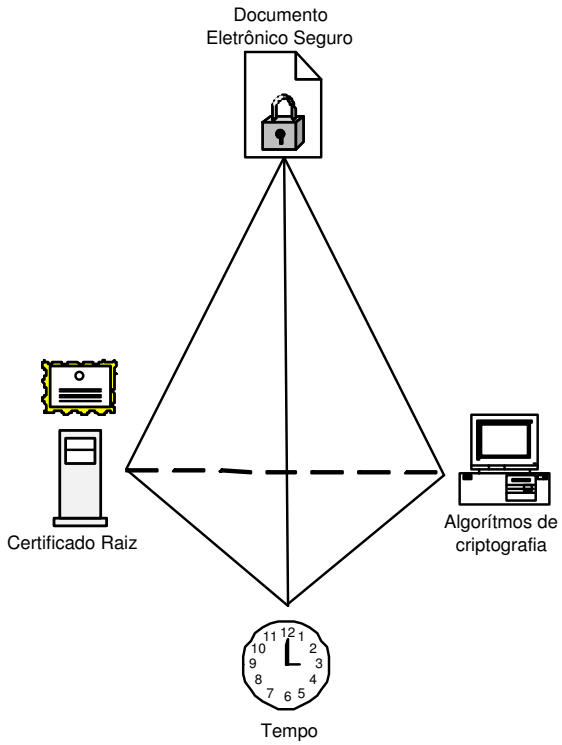

Figura 1.1: Base de um documento eletrônico seguro.

Dentre os requisitos de segurança, a tempestividade é talvez o que menos tem despertado a atenção dos pesquisadores. Entretanto, este requisito é muito importante, visto que a tempestividade associa uma determinada ação a um ponto no tempo, ou seja, cria uma âncora temporal, garantindo que a partir da datação do documento ele não possa mais ser modificado sem que haja vestígios.

Sabe-se contudo que um documento eletrônico, para ser considerado seguro, deve possuir uma âncora temporal confiável (DIAS; CUSTÓDIO; DEMÉTRIO, 2003). Associado a qualquer âncora temporal deve existir um relógio que esteja sincronizado com a hora oficial de uma nação, para que esta âncora seja considerada confiável. Todavia, a distribuição da informação de data e hora e a sincronização dos relógios dos

equipamentos não é uma tarefa trivial. Além disso, o carimbo do tempo preciso em um documento faria sentido somente se a datação de todos os documentos fosse realizada apenas por entidades que possuem o tempo oficial e confiável, como o Observatório Nacional no Brasil. Como isto não é viável na prática, devido a uma possível sobrecarga das entidades que possuem o tempo oficial e devido a limitações físicas, o tempo é distribuído para servidores de tempo, os quais o distribuirão tentando minimizar perdas e defasagens perante o horário oficial. Para tanto, há a necessidade que haja um sincronismo entre os servidores de tempo, para que estes conservem a menor diferença possível entre seus relógios e o horário oficial.

Outro detalhe importante a ser considerado é como o horário confiável do relógio do servidor de tempo será agregado a um documento eletrônico. A Autoridade de Datação (AD) é responsável por associar o tempo do mundo real ao documento virtual.

Além do tempo absoluto ou datação absoluta, que garante o momento da criação de um documento "carimbando" no mesmo o tempo proveniente de uma fonte de tempo confiável, há também a forma de datação onde um determinado documento que está sendo datado é relacionado ao documento que foi datado anteriormente, de forma que, passado um determinado período de tempo, exista uma cadeia onde cada documento está encadeado ao seu antecessor e ao seu sucessor. Este método é chamado de datação relativa. Estas formas de datação, absoluta e relativa, reunidas em um mesmo método de datação, chamado de datação híbrida, fazem com que o número de possibilidades de fraudes seja minimizado. Os métodos de datação absoluta, relativa e híbrida são melhor detalhados no capítulo 2. Algumas ADs utilizam tanto a datação absoluta quanto a relativa ou híbrida para fixar a âncora temporal. Para este trabalho, dá-se ênfase às ADs que utilizam a datação híbrida.

Na datação absoluta há a necessidade do sincronismo do relógio da AD com uma fonte de tempo confiável, pois este método de datação necessita do tempo com uma precisão consideravelmente boa (a precisão depende da finalidade dos documentos). Para tanto, protocolos que desempenhem este papel são requeridos. Os principais protocolos de sincronismo estão detalhados no capítulo 3.

Até o presente trabalho, tratou-se de uma AD isoladamente[3], seus métodos e necessidades. Mas, na prática, uma AD pode não ser suficiente para atender a demanda de pedidos de protocolação de uma instituição. Outro problema é a comparação temporal de documentos datados em diferentes ADs. Caso a AD utilize datação absoluta, nada garante que o relógio das ADs que dataram os documentos a serem comparados estão corretos, e se a AD utiliza datação relativa, nada se pode afirmar em relação ao tempo de documentos datados em diferentes ADs. Assim, pode-se pensar em uma "rede de ADs". Para tanto, há a necessidade da concepção de uma infra-estrutura que abranja todos os aspectos relevantes ao funcionamento de uma "rede de ADs", como os elementos que fazem parte da infra-estrutura, os protocolos envolvidos, a segurança, o gerenciamento e a auditoria.

Fazendo parte da infra-estrutura estão os clientes de ADs (que solicitam a datação de documentos eletrônicos), as ADs (datando documentos com os métodos de datação relativa e/ou absoluta, sincronizando seus relógios com fontes ou servidores de tempo confiáveis e sincronizando suas cadeias de datação entre si em caso de datação relativa), servidores ou fontes de tempo (que disponibilizam tempo aos seus clientes) e clientes de tempo (que são todas as entidades que requisitam tempo para a fonte de tempo). A Figura 1.2 representa a infra-estrutura de maneira geral.

Para que o gerenciamento e a auditoria da infra-estrutura de protocolação digital possam acontecer, os elementos devem estar devidamente identificados e autenticados, principalmente os servidores e os clientes de tempo, já que a auditoria irá se preocupar em rastrear e auditar o tempo dos clientes de tempo (os principais clientes de tempo são as ADs). A autenticação deve ser feita através de Infra-Estrutura de Chaves Públicas[4] (ICP), ou seja, através de certificados digitais emitidos por Autoridades Certificadoras (ACs), pois agregam maior confiança quando comparados com os certificados que não são emitidos por ACs, como os certificados auto-assinados, por exemplo.

Este modelo de infra-estrutura, até o presente trabalho, ainda não foi proposto ou explorado. Há uma dissertação de mestrado de Roos (1999), o qual foi ori-

[3]Os trabalhos que tratam de datação de documentos eletrônicos estão relacionados na seção 1.4.

[4]Vide glossário (apêndice A).

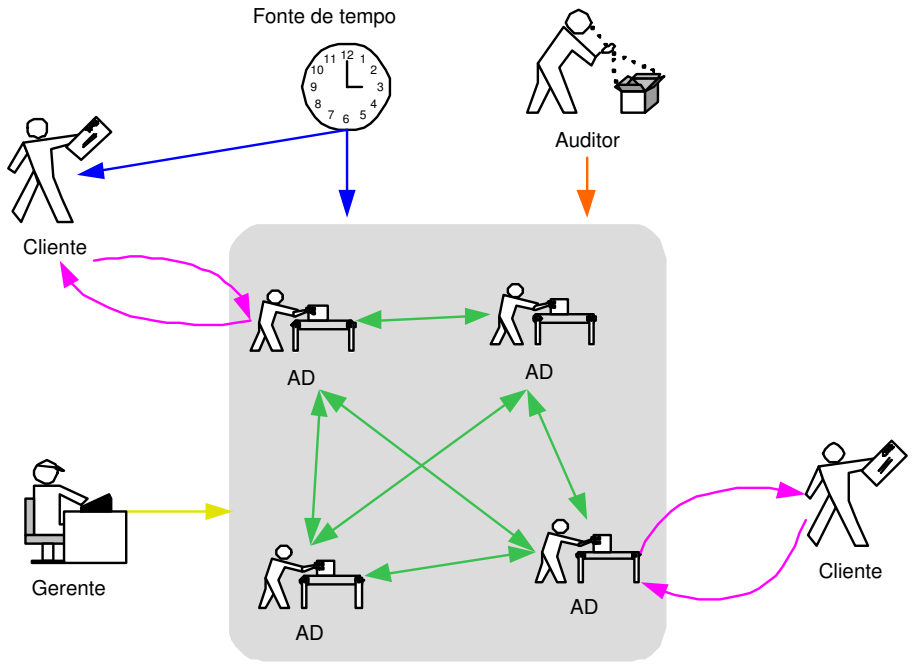

Figura 1.2: Infra-estrutura de protocolação digital distribuída de doc. eletrônicos.

entado por Helger Lipmaa, que possui um capítulo sobre acúmulo de “notários” (como o autor se refere para as ADs). A infra-estrutura proposta por ele utiliza uma hierarquia onde há um “notário superior” que assina resumos emitidos por outros notários. Ele trabalha com conceitos de rodadas, sendo que todos os pedidos de protocolação de uma rodada, representado por um intervalo de tempo, são utilizados para gerar um resumo. Depois do resumo assinado pelo notário superior, cada “notário inferior” na hierarquia envia uma prova para os clientes que submeteram pedidos de datação em uma determinada rodada de que seus dados foram utilizados para fazer um resumo e que este foi assinado pelo notário superior.

A proposta deste trabalho é definir uma infra-estrutura de protocolação digital, incluindo a definição dos seus elementos, protocolos de comunicação, gerência e auditoria. A proposta da infra-estrutura é explanada no capítulo 6. É utilizado o Protocolo de Tempo em Rede versão 4 (*Network Time Protocol* - NTP) para o sincronismo dos relógios, mas é agregado a este protocolo a funcionalidade da autenticação com certificados digitais assinados por ACs externas (visto que esta versão do NTP utiliza certificados

digitais auto-assinados emitidos pelo próprio NTP) tanto para os clientes quanto para os servidores de tempo (na atual implementação do NTP apenas o servidor é autenticado). É também acrescentado um auditor que faz a auditoria de tempo nos relógios das ADs e trabalha em conjunto com o protocolo NTP. Estas adaptações não afetam o funcionamento normal do NTP. As funcionalidades adicionadas no protocolo NTP são detalhadas no capítulo 5.

1.1 Definição do Problema

Para que uma instituição que utiliza documentos eletrônicos em suas transações possa fazer uso dos mesmos como meio de provas jurídicas, segundo Marcacini (1999), é necessário que eles possuam algumas características presentes nos documentos tradicionais, ou seja, que estes possuam autoria identificável e que o documento não possa ser alterado de forma imperceptível. De semelhante maneira como nos documentos tradicionais, os documentos eletrônicos devem possuir um carimbo de tempo. Mas como os documentos eletrônicos são uma seqüência de bits e são independentes de continentes, há a necessidade de métodos que garantam que os documentos eletrônicos sejam datados de maneira segura e confiável, sendo que qualquer alteração em sua data seja identificada.

Neste sentido, muitos estudos estão sendo realizados. No LabSEC (Laboratório de Segurança em Computação da UFSC) (LABSEC, 2003) foi finalizada a dissertação de mestrado de Pasqual (2002) que descreve os métodos de datação de documentos eletrônicos existentes e propõe um novo método de datação. Há também esta dissertação de mestrado e a de Costa (2003), ambas tratando de datações de documentos eletrônicos, mas com enfoques diferentes. Estes estudos são importantes para a área, visto que o governo brasileiro está incentivando o uso de documentos eletrônicos em repartições públicas através do Governo Eletrônico (BRASIL, 2001).

Estudando-se as soluções encontradas para resolver o problema da datação eletrônica, uma questão surgiu em virtude da concretização destas soluções: o domínio da maioria dos estudos na área de datação de documentos eletrônicos trata de uma

Autoridade de Datação isolada. Assim, surgiu a necessidade de criar uma infra-estrutura para dar sustentação à toda a estrutura de protocolação, ou melhor, uma espécie de “rede” de protocolação digital de documentos eletrônicos. Entretanto, para este tipo de topologia ser viável, devem ser considerados alguns aspectos como a rastreabilidade e a autenticação dos elementos da rede, a gerência, protocolos envolvidos e uma forma de auditoria, para que se possa estabelecer confiança nos tempos transmitidos e recebidos.

O decreto 4.264 de 10 de junho de 2002 (BRASIL, 2002), reafirma que o Observatório Nacional (ON) é responsável pela disseminação e geração da hora legal brasileira. Assim, o ON é a fonte oficial da hora brasileira. É de interesse do ON que seu tempo esteja sendo transmitido corretamente para seus clientes devidamente autenticados perante ele e que o ON possa auditar o tempo que seus clientes estejam utilizando, inclusive de Autoridades de Datação, comparando se está de acordo com o tempo transmitido pelo próprio ON. Para que o tempo seja transmitido do ON para outras instituições ou para pessoas em particular, há a necessidade de protocolos que transmitam o horário sem defasagens ou perda da precisão do tempo, visto que se está trabalhando com tempo e dependendo da aplicação, a precisão é extremamente importante, como por exemplo em transações bancárias, em autoridades certificadoras, em transações de comércio eletrônico e qualquer outro tipo de aplicação via Internet que possua algum tipo de prazo.

Devido aos fatos mencionados acima, o ON mostra interesse em fazer uma parceria com o LabSEC para a pesquisa e o desenvolvimento de um sistema de gerência, auditoria e rastreabilidade do tempo. Assim, o ON poderá rastrear e auditar o tempo de seus clientes, incluindo Autoridades de Datação, investigando se estão utilizando realmente o tempo que o ON forneceu. A Figura 1.3 mostra a tarefa do ON.

Assim, este trabalho propõe definir uma infra-estrutura de protocolação digital, descrevendo os elementos envolvidos e os protocolos utilizados. Propõe-se também definir uma forma de gerência, rastreabilidade, identificando os elementos envolvidos com a recepção e fornecimento do tempo, e auditoria do tempo que está tramitando na infra-estrutura.

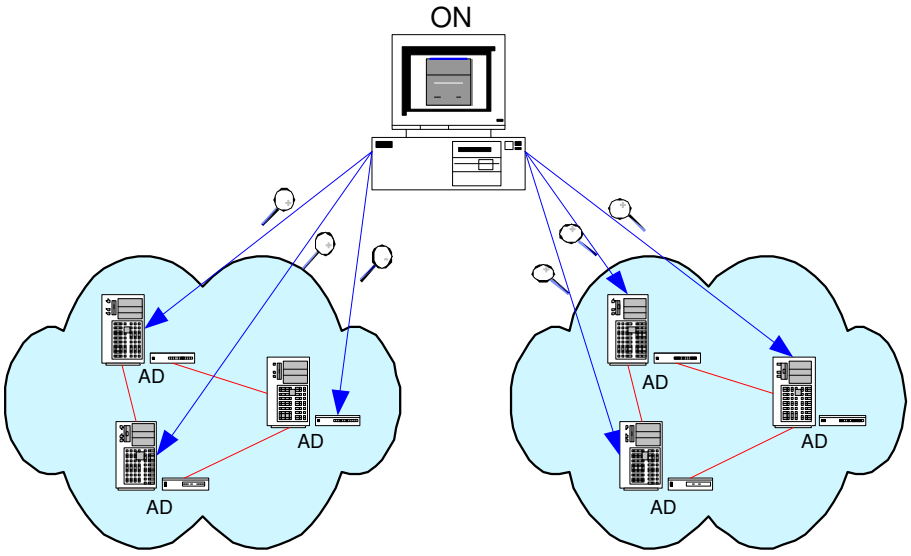

Figura 1.3: Auditoria do ON sobre as ADs.

1.2 Objetivos

1.2.1 Objetivo Geral

Propor uma infra-estrutura para a protocolação digital de documentos eletrônicos.

1.2.2 Objetivos Específicos

- Definir os componentes e protocolos necessários para uma infra-estrutura de protocolação digital de documentos eletrônicos;
- Propor e implementar uma forma de rastreabilidade e auditoria do tempo nos membros pertencentes à infra-estrutura;
- Fazer a validação formal da auditoria e rastreabilidade acima citadas;
- Definir o formato do recibo de um documento datado em uma AD pertencente à infra-estrutura proposta, incluindo a datação relativa e a protocolação cruzada;
- Definir um protocolo de gerência para a infra-estrutura.

1.3 Materiais e Métodos

Esta pesquisa tem caráter teórico e prático, como mostram os objetivos supracitados.

O trabalho está fundamentado em artigos, livros e diversos tipos de documentos relacionados com a área de criptografia e datação de documentos eletrônicos. Utiliza-se o ambiente do LabSEC (LABSEC, 2003) para a realização dos experimentos, implementação e estudos envolvidos. Há disponível uma rede interna com servidores de Web e servidores de tempo onde os experimentos foram realizados.

O trabalho foi desenvolvido da seguinte maneira: Na primeira etapa realizou-se uma pesquisa bibliográfica em documentos relacionados à área de segurança da informação, mais especificamente relacionados com o manejo do tempo em ambientes virtuais. Após esta etapa, analisou-se e definiu-se a infra-estrutura de protocolação digital de documentos eletrônicos e implementou-se o que foi necessário, com o apoio de alunos da graduação de ciência da computação da UFSC. No documento de dissertação são descritos os conceitos e conhecimentos adquiridos com a pesquisa e a implementação, além da proposta do trabalho, atendendo assim aos objetivos pretendidos.

1.4 Trabalhos Correlacionados

Há dez anos a datação de documentos eletrônicos era ainda uma área um tanto obscura e precisava ainda ser estudada. Após a publicação de Haber e Stornetta (1991) propondo o método de datação relativa do encadeamento linear[5], esta área passou a ser mais conhecida e alguns grupos de pesquisa se interessaram por ela.

A primeira publicação que tratou diretamente de requisitos de segurança da Autoridade de Datação (AD) foi o artigo de Buldas (1998). Neste artigo é proposto uma terceira entidade confiável para realizar a datação em documentos eletrônicos e formularam o conceito de datação relativa, permitindo a auditoria da datação. Este artigo também define um novo método que não é baseado em encadeamento seqüencial, intro-

[5] Vide glossário (apêndice A).

duzindo assim um novo termo chamado rodadas, sendo este método mais eficiente do que um método relativo que possui uma forma seqüencial de encadeamento, como por exemplo o método do encadeamento linear. A inovação deste método se dá na busca pelo resumo de um documento que tenha sido datado, o tempo para a realização desta tarefa foi reduzido.

Um novo método baseado em encadeamento binário foi proposto por Buldas e Laud (1998). Este artigo formaliza o esquema de encadeamento binário, o qual é formado por um grafo acíclico dos recibos de datação. A vantagem deste método é que vários caminhos podem ser percorridos para busca de um mesmo recibo.

O artigo de Buldas, Lipmaa e Schoenmakers (2000) está relacionado com o artigo anterior de Buldas e prova que o grafo acíclico é desnecessário para a garantia da auditoria da Autoridade de Datação. Neste mesmo artigo, outros tipos de grafos foram propostos com a intenção de diminuir o tempo de busca de um recibo no encadeamento.

Em 2002, foi finalizada a dissertação de mestrado de Everton Schonardie Pasqual, membro do LabSec da UFSC (PASQUAL, 2002). Esta dissertação propõe outro método de datação relativa, baseado nos métodos anteriores, mas adicionando a eles ainda novos conceitos para agilizar o processo de busca dos recibos. Também foram publicados artigos explanando este novo método de datação de documentos eletrônicos, sendo os autores dos artigos Pasqual, Dias e Custódio (2002a, 2002b). Adriana Elissa Notoya publicou sua dissertação em 2002 (NOTOYA, 2002) na UFSC; seu trabalho trata da validade do documento eletrônico por tempo indeterminado. Além destes trabalhos, há também o trabalho de Vanessa Costa que trata dos aspectos relacionados à confiabilidade da protocolação (COSTA, 2003).

Outros projetos também estão sendo desenvolvidos na área de datação eletrônica, como o projeto Cuculus[6] de um grupo de estonianos, bem como projetos na Alemanha, França, Estados Unidos, Austrália e Finlândia. Maiores informações sobre projetos e empresas que atuam na área de datação de documentos eletrônicos podem ser

[6]http://www.tcs.hut.fi/ helger/cuculus/

encontradas na página de Lipmaa (2003)[7] .

1.5 Justificativa e Motivação

Devido à importância da datação em documentos eletrônicos, como já descrito na introdução deste capítulo, a área de protocolação digital está sendo pesquisada há mais ou menos uma década. Neste tempo, surgiram várias publicações relacionadas aos métodos e formas de datação, como mostrado na seção 1.4. Entretanto, a iniciativa de criar uma infra-estrutura para a protocolação digital com Autoridades de Datação utilizando métodos de datação absoluta e relativa ainda não foi explorada. Mesmo detalhes de implementação de uma infra-estrutura como gerência, auditoria entre outros aspectos abordados neste trabalho também não foram pesquisados.

Uma rede de ADs é muito importante para grandes empresas que estão adotando a tecnologia da virtualização de documentos como suporte para suas transações, visto que para uma demanda grande de datações de documentos, apenas uma AD pode não ser suficiente. Atualmente, as empresas que estão utilizando ADs ainda possuem equipamentos isolados, sem ligação alguma entre eles.

Além disso, o LabSEC possui um grande projeto chamado Cartório Virtual. Este projeto está subdividido em várias partes, como certificação digital, emissão de registros públicos, datação de documentos eletrônicos, entre outros. O presente trabalho está sob o escopo do subprojeto de datação de documentos eletrônicos, onde também está o projeto da Protocoladora Digital de Documentos Eletrônicos (PDDE).

Outra motivação para este trabalho é a proposta do governo brasileiro de um Governo Eletrônico (BRASIL, 2001). Este projeto incentiva a utilização de documentos eletrônicos em, principalmente, órgãos públicos. Algumas das atividades realizadas pelo governo já podem ser feitas através dos meios digitais, como a declaração do imposto de renda. Assim, o governo está incentivando pesquisas na área de segurança da informação.

[7] Pesquisador da área de datação de documentos eletrônicos.

1.6 Organização do Texto

Os primeiros capítulos deste trabalho apresentam a revisão bibliográfica necessária para a realização do mesmo.

Para que se fale de documentos eletrônicos é necessário uma explicação do que é um documento eletrônico, suas diferenças em relação aos documentos em papel e as formas e os métodos de datação de documentos eletrônicos. Estas definições são encontradas no capítulo 2. O capítulo 3 descreve os principais protocolos utilizados para o sincronismo de relógios, ou melhor, como ter o tempo preciso de um relógio atômico, por exemplo, em um computador. Dentre os protocolos explanados no capítulo 3, um deles foi escolhido para fazer parte da infra-estrutura de protocolação digital: o Protocolo de Tempo em Rede (*Network Time Protocol* - NTP). A descrição detalhada deste protocolo e a razão deste ser escolhido para fazer parte da infra-estrutura encontra-se no capítulo 4.

Depois de abordados todos os conceitos necessários para a realização do trabalho, inicia-se os capítulos de contribuição.

Para que o protocolo NTP seja utilizado na infra-estrutura proposta garantindo toda a segurança necessária, há a necessidade de acrescentar algumas funcionalidades neste protocolo, descritas no capítulo 5. Tendo reunido todos os componentes, descreve-se então a infra-estrutura no capítulo 6, que dá o nome ao trabalho.

O capítulo 7 destina-se às considerações finais e aos trabalhos futuros.

Um guia rápido dos principais termos utilizados neste trabalho é descrito no apêndice A. A implementação das funcionalidades acrescentadas no NTP é tratada com detalhes no apêndice B. Na última parte do trabalho, apêndice C, estão os artigos publicados, na íntegra, durante o período de mestrado.

Capítulo 2

Datação de Documentos Eletrônicos

2.1 Introdução

Com as facilidades que computadores e Internet proporcionam, estão surgindo maneiras de automatizar e virtualizar as tarefas do dia a dia. Para que estas tarefas tenham a mesma eficiência que possuem no mundo real, as tarefas realizadas no mundo eletrônico devem possuir as mesmas garantias que as do mundo real oferecem, como autenticidade, integridade e tempestividade. Para garantir a autenticidade e integridade, são utilizados tecnologias como as funções resumo (*hash*) e assinaturas digitais (STINSON, 1995; MENEZES; OORSCHOT; VANSTONE, 1996). Para a tempestividade, utiliza-se métodos que possibilitem associar uma âncora temporal - amarrar alguma ação a um determinado ponto no tempo - ao documento eletrônico, sendo que esta associação deve ser confiável e que não haja maneira de ser violada sem que algo seja percebido. A âncora temporal do documento eletrônico (*timestamp*) é dada por uma ou mais entidades que carimbam o tempo no documento eletrônico através de métodos de datação. Quando a datação é feita por apenas uma entidade, esta entidade é chamada de Autoridade de Datação (AD).

Há três formas de datação de documentos eletrônicos utilizados por ADs: a absoluta, a relativa e a híbrida. Dentro do conceito da datação relativa, existem várias técnicas de protocolação de documentos eletrônicos, entre elas o método de enca-

deamento linear, o método da árvore e o método da árvore sincronizada. Estas técnicas serão melhor detalhadas nas seções seguintes.

O presente capítulo é dividido nas seguintes seções: na seção 2.2 são explanados os documentos tradicionais e os documentos eletrônicos, mostrando suas características, vantagens e desvantagens. Na seção 2.3 são descritas as formas de datação absoluta, relativa e híbrida. Na seção 2.4 são explicados com detalhes os métodos de datação relativa de documentos eletrônicos mais utilizados. Na seção 2.5 o capítulo é concluído.

2.2 Documentos

Com o crescimento do uso de documentos eletrônicos o próprio conceito de documento vem mudando ao longo do tempo. O substantivo “documento” não se refere mais a apenas documentos em papel, mas sim a todo tipo de registro que possa ser consultado posteriormente, incluindo assim também os documentos eletrônicos.

Em se tratando da eficácia jurídica dos documentos, de acordo com Bortoli (2002), um documento é válido juridicamente levando em conta a relevância jurídica do escrito, ou seja, o significado jurídico da expressão contida nele. É necessário que a expressão do pensamento nele contido tenha a possibilidade de gerar conseqüência no plano jurídico.

Para que um documento seja utilizado como prova jurídica, segundo Bortoli (2002), este deve atender alguns requisitos básicos:

- **Integridade**: o conteúdo do documento não pode ser alterado sem que seja percebido;
- **Autenticidade**: deve ser comprovada a autoria do documento;
- **Tempestividade**: o documento deve ser datado;

Como se pode notar, os ítens acima são independentes de continente, ou melhor, são válidos tanto para documentos tradicionais (como os documentos em papel) quanto para qualquer outro tipo de documento. Para documentos eletrônicos, por

exemplo, estes ítens devem ser garantidos com um alto nível de segurança e caso sejam violados, deve haver uma forma de perceber a violação, tal qual nos documentos em papel.

Nas subseções seguintes, serão melhor detalhados os documentos tradicionais e os documentos eletrônicos.

2.2.1 Documentos Tradicionais

Os documentos tradicionais são, geralmente, apresentados em papel. Mas o papel não é o único continente utilizado. Antigas civilizações também utilizam documentos mesmo antes do papel ser descoberto. Os documentos eram escritos em pedras, madeiras, papiros, pele de animais, cerâmica, entre outros.

Com o passar do tempo, os documentos evoluíram, passaram a ser escritos em papel e posteriormente também em meios eletrônicos.

Mesmo com a crescente utilização dos documentos eletrônicos, os documentos tradicionais ainda são muito utilizados. Existe, por parte da população em geral, um certo preconceito contra documentos eletrônicos, sua validade e segurança ainda são questionadas e devido a este fato, documentos em papel são ainda muito utilizados. Isto se deve à falta de informação sobre o assunto da maioria da população que utiliza meios eletrônicos e por não ser tão fácil de perceber certas características no documento eletrônico como no documento de papel, como a assinatura, por exemplo. Há a necessidade de ter e confiar em um ou mais *softwares* para que certas características tão facilmente percebidas em documentos de papel sejam percebidas ou visualizadas em documentos eletrônicos.

No documento em papel, na maioria dos casos, é fácil saber se o documento foi rasurado, modificado ou qual é o original e qual é a cópia. Já para documentos eletrônicos, esta distinção não é tão fácil senão impossível. Esta é uma das grandes vantagens dos documentos tradicionais. Além desta vantagem, o documento tradicional é algo palpável, o que dá mais segurança e confiabilidade a uma pessoa que não é muito familiarizada com a tecnologia digital.

Uma das grandes características dos documentos tradicionais é que a informação contida no documento é indissociável do seu suporte, conteúdo e continente se integram como um todo (VEIGA, 2002). Para o documento tradicional, o documento em um todo é constituído pela informação e pelo suporte que contém esta informação (geralmente é o papel). Esta característica não é percebida nos documentos eletrônicos, já que estes não possuem continente fixo, a mesma informação pode possuir diversos continentes sem que o conteúdo seja alterado.

Apesar destas características, os documentos tradicionais possuem algumas desvantagens em relação aos documentos eletrônicos, como por exemplo o volume ocupado por documentos em papel. O Brasil, hoje, possui muitos problemas com arquivos públicos que estão cheios de documentos que não podem ser extraviados. Outra desvantagem do documento tradicional é a menor praticidade em sua transmissão, visto que existe a necessidade de um aparelho de fax (que não transmite o documento original e sim faz uma cópia deste e a transmite), dos correios ou de uma pessoa que entregue o documento para o destinatário.

Hoje em dia, todas as transações e atividades humanas (principalmente jurídicas) são baseadas em documentos sendo, na grande maioria, documentos em papel.

2.2.2 Documentos eletrônicos

Documentos eletrônicos são informações armazenadas em meios magnéticos, sendo dissociados de continente. Continente e conteúdo não se confundem. Um documento eletrônico pode ser tratado como uma seqüencia de bits, ou seja, uma seqüência de 0's e 1's. Todos os tipos de dados que um documento pode conter como números, fotos, texto, entre outros, no documento eletrônico, são representados de uma só maneira, na forma binária.

A disseminação dos documentos eletrônicos vem crescendo com o passar dos anos e evoluindo rapidamente. Ainda existe um tabu em relação à utilização de documentos eletrônicos em certas transações e atividades humanas, devido ao fato de que muitas pessoas ainda precisam ter algo palpável para sentirem-se seguras. Esta é uma

desvantagem dos documentos eletrônicos, a resistência que eles ainda possuem perante uma parte da população, principalmente no Brasil.

Em se tratando de documentos eletrônicos, não há distinção do documento “original” e suas “cópias”, todos podem ser considerados originais. Outra característica dos documentos eletrônicos que se pode considerar desvantagem é que qualquer tipo de alteração não deixará rastros, como acontece no documento tradicional, a não ser que seja aplicada alguma medida de prevenção, como a função resumo ou a assinatura digital.

No Brasil, a idéia de um Governo Eletrônico foi proposta em agosto de 2001 (BRASIL, 2001) pelo ministério do planejamento, orçamento e gestão. Esta proposta incentiva o uso de documentos eletrônicos para o uso federal e incentiva que uma quantidade maior de transações que hoje só acontecem com documentos tradicionais seja feita com documentos eletrônicos. Hoje em dia, alguns serviços aos cidadãos brasileiros já são disponibilizados na Internet, como por exemplo a declaração do imposto de renda, emissões de certidão de pagamento de impostos, acompanhamento de processos judiciais, programas de ensino à distância, entre tantos outros.

Esta proposta de governo eletrônico prevê os seguintes ítens (BRASIL, 2001):

- Oferta na Internet de todos os serviços prestados ao cidadão, com melhoria dos padrões de atendimento, redução de custos e facilidade de acesso;

- Ampliar o acesso à informação pelo cidadão, em formatos adequados, por meio da Internet;

- Promover a convergência entre sistemas de informação, redes e bancos de dados governamentais para permitir o intercâmbio de informações e a agilização de procedimentos;

- Implantar uma infra-estrutura avançada de comunicações e de serviços, com padrões adequados de segurança e serviços, além de alto desempenho;

- Utilizar o poder de compra do Governo Federal para a obtenção de custos menores e a otimização do uso de redes de comunicação;

- Estimular o acesso à Internet, em especial por meio de pontos de acesso abrigados em instituições públicas ou comunitárias;

- Concorrer para o fortalecimento da competitividade sistêmica da economia.

Este é um grande salto para a utilização de documentos eletrônicos no Brasil, já que o governo federal propõe a utilização deste tipo de documento para os serviços prestados pelo próprio governo.

2.3 Formas de datação

Como já mencionado anteriormente, os documentos eletrônicos devem satisfazer três critérios: integridade, autenticidade e tempestividade. A integridade é garantida por funções de sentido único, como funções resumo (*hash*). A autenticidade é atendida por assinatura digital. Já o requisito de tempestividade é atendido pela datação, protocolação digital ou carimbo de tempo[1], os quais agregam uma âncora temporal ao documento eletrônico. A protocolação pode ser feita de duas maneiras: por métodos baseados em confiança distribuída (BENALOH; MARE, 1991, 1994) ou por métodos baseados em uma Autoridade de Datação (AD).

As técnicas baseadas em AD supõem que esta é confiável, ou seja, que ela permanece imparcial ao processo. Já as técnicas baseadas em confiança distribuída baseiam-se no fato da operação ser feita por um grupo, ou melhor, um grupo específico assina e data o documento de maneira que convença o verificador da legitimidade da ação, pois seria mais difícil corromper todos os membros do grupo para ocorrer a fraude (JUST, 1998).

Segundo Haber e Stornetta (1991), um método de datação deve atender aos seguintes requisitos:

[1] Vide glossário (apêndice A).

- **Privacidade**: O método deve manter a privacidade do documento de forma que apenas o cliente saiba qual o documento que está sendo datado;
- **Canal de Comunicação e Armazenamento**: Independentemente do tamanho do documento, deve ser fácil de datá-lo e armazená-lo;
- **Erro de Comunicação**: O arquivo não deve ser corrompido ou adulterado em sua transmissão ou em seu armazenamento;
- **Confiança**: Deve-se garantir que o documento será datado com a data e hora corretas.

Dados estes requisitos, segundo Pasqual, Dias e Custódio (2002a), o método de datação utilizando uma Autoridade de Datação é o mais indicado para satisfazê-los, pois com a datação baseada em confiança distribuída, o documento a ser datado deve percorrer diversos caminhos e há maior tendência de ocorrer erros nesta tramitação.

Deve-se levar em conta também como o documento eletrônico irá receber o carimbo de tempo. Com o método utilizando AD, fica mais fácil e prático datar o documento, pois apenas uma entidade irá datar e manusear o documento. A datação no documento pode ser feita na forma absoluta, relativa ou híbrida, que serão explicados a seguir.

2.3.1 Datação absoluta

A datação absoluta prevê a total confiança na Autoridade de Datação e na fonte que está fornecendo o tempo para os carimbos de tempo. Este tipo de datação consiste em anexar no resumo do documento o horário (do mundo real) da chegada do documento na AD, que deve possuir uma fonte de tempo confiável disponível para que seu carimbo de tempo seja confiável.

Este tipo de datação tem alguns problemas pois se a AD for maliciosa, ou se algum erro ocorrer com a mesma ou com a fonte de tempo, então nada impede que a AD emita carimbos de tempo errados, com data e hora diferentes do horário correto no momento do carimbo.

A confiabilidade do método está na confiança na AD e na fonte de tempo conectada a ela.

2.3.2 Datação relativa

A Datação Relativa não trabalha com o tempo absoluto, todavia trata de informações que dizem se um documento foi protocolado antes ou depois de outro. Os resumos dos documentos que vão chegando para serem protocolados são encadeados no resumo anterior e assim por diante, formando assim uma cadeia. Esta cadeia é armazenada no banco de dados da AD e em alguns métodos de datação relativa, o recibo de datação também contém informações do encadeamento que permitem reconstruir a cadeia. Na datação relativa, a total confiança na Autoridade de Datação não é necessária, pois pode-se verificar a seqüência de protocolações e descobrir se algo está errado.

O encadeamento L_n é formado pela função resumo aplicada nos resumos dos documentos H_n e nos L's anteriores, onde X_n é o documento a ser datado e $H_n = H(X_n)$, satisfazendo à Equação recursiva:

$$L_n = H(H_n, L_{n-1}, \ldots, L_{m-n}) \tag{2.1}$$

onde m é o tamanho da cadeia de encadeamento que também pode ser chamado de tamanho da cadeia de verificação. L_n é o encadeamento do n-ésimo documento. Pode-se ressaltar aqui que o documento que possui o encadeamento L_{n-1} não foi necessariamente criado antes de um documento que possui o encadeamento L_n, mas sim que este documento foi enviado para a AD antes que o outro.

O protocolo para datar um documento eletrônico utilizando datação relativa consiste nos seguintes passos (LOMBARDI; LIPMAA, 1998):

1. Antes de assinar o documento X, Alice (interessada na datação de um documento eletrônico) calcula a função resumo do documento e o envia para a AD para ser datado;

2. Após o retorno do recibo do resumo do documento datado, Alice anexa o recibo

$L(H)$ no próprio documento, assina-o e envia-o para a AD, obtendo o recibo $L(X')$ da assinatura $sig_A = (L(H), X)$

3. Como existe uma dependência de caminho único entre $L(H)$, X e $L(X')$ o verificador pode concluir que a assinatura foi feita entre o tempo da criação de $L(H)$ e de $L(X')$ respectivamente.

Assim, pode-se determinar, aproximadamente quando o documento foi assinado. Podemos ver esse procedimento na Figura 2.1.

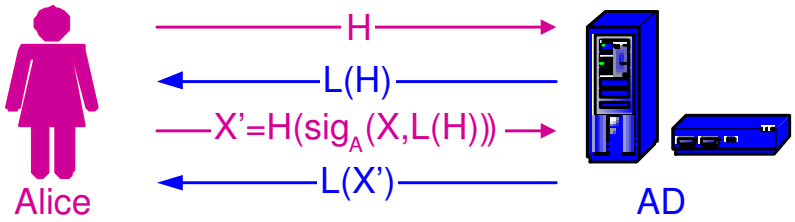

Figura 2.1: Datação de um documento eletrônico utilizando datação relativa.

Para a maioria das aplicações a datação relativa pode ser considerada mais eficaz, devido à garantia de uma maior segurança, visto que torna-se mais difícil fraudar o encadeamento. A desvantagem deste tipo de datação perante a datação absoluta é que a datação relativa necessita de maiores recursos computacionais e a sua verificação é um pouco mais dispendiosa.

2.3.3 Datação híbrida

A datação híbrida é a utilização da datação absoluta e relativa simultaneamente, ou seja, anexar a data e hora absoluta ao documento e também armazenar a seqüência dos documentos que vão sendo protocolados, encadeando um documento com seu antecessor sucessivamente.

Esta forma de datação é a mais confiável dentre as três aqui expostas pois, além de carimbar a hora absoluta no recibo, este também contém informações sobre o encadeamento do documento, permitindo que a cadeia seja reconstruída, possibilitando

a verificação da idoneidade do encadeamento ou a comparação entre dois documentos datados em uma mesma AD baseando-se apenas no encadeamento.

2.4 Métodos de datação relativa

Com o estudo da datação relativa, os métodos de datação derivados da datação relativa foram surgindo, como o método de encadeamento linear, o da árvore e o da árvore sincronizada que serão melhor explicadas a seguir.

2.4.1 Encadeamento Linear

No método do encadeamento linear não há a necessidade de confiar cegamente na AD, pois os resumos dos documentos são encadeados entre si por uma função H, conceito derivado da forma de datação relativa.

No método do encadeamento linear o recibo s de um documento H_n é definido como (BULDAS, 1998):

$$s = sig_{AD}(n, t_n, ID_n, H_n, L_n) \tag{2.2}$$

onde sig_{AD} é a assinatura da AD, t_n é a data e hora corrente, ID_n é o indicador do n-ésimo documento e L_n é a informação do *link* definido como:

$$L_n = (t_{n-1}, ID_{n-1}, H_{n-1}, H(L_{n-1})) \tag{2.3}$$

onde t_{n-1} e ID_{n-1} são, respectivamente, data e hora e o identificador do documento anterior, H_{n-1} é o resumo do documento anterior e $H(L_{n-1})$ é o resumo do *link* anterior. A Figura 2.2 representa a Equação da criação dos encadeamentos.

O L_0 é um número aleatório criado pela AD para iniciar a cadeia, sendo considerado o *link* zero. Quando o resumo do primeiro documento a ser datado chega para a AD, esta calcula o primeiro *link* da cadeia de acordo com a Equação 2.3, assina o recibo contendo as informações descritas na Equação 2.2 e envia este recibo para o cliente. Baseando-se em sua base de dados, a AD terá condições de saber a seqüencia dos documentos submetidos para datação.

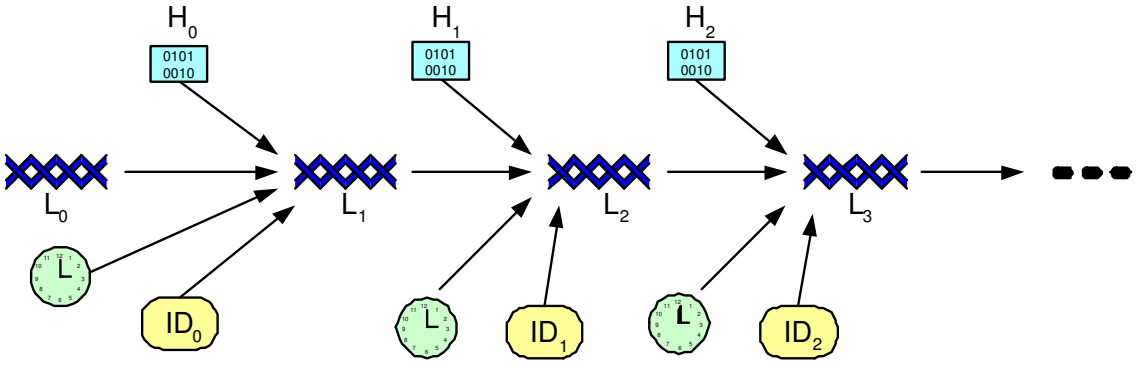

Figura 2.2: Construção do *link*.

2.4.2 Árvore

Há graves problemas com a implementação prática do método do encadeamento linear, segundo Buldas (1998). O processo de comparação temporal entre dois documentos protocolados e armazenados no encadeamento pode ser muito dispendioso. Uma simples verificação pode ser muito mais custosa que a criação da própria cadeia. Há soluções para este problema, as quais estão explicadas em (BENALOH; MARE, 1991; HABER; STORNETTA, 1991).

Além das soluções encontradas por Haber e Benaloh, uma outra solução foi criada com o objetivo de agilizar a verificação e impor maior confiança ao método de encadeamento linear. A proposta geral deste método é criar unidades de tempo, denominadas rodadas (BAYER; HABER; STORNETTA, 1992). É baseada em grafos e foi chamada de Método da Árvore.

O tamanho de uma rodada pode ser definido como a quantidade máxima de solicitações ou como um intervalo de tempo. O principal objetivo deste método é melhorar o desempenho na verificação de dois documentos protocolados, pois eventualmente, o verificador poderá verificar documentos em apenas uma das rodadas, o que irá reduzir bastante o tempo de busca.

O *link* de rodada R_r para a rodada r é o resumo (*hash*) acumulado do *link* de rodada R_{r-1} da rodada $r-1$ e de todos os documentos submetidos à AD durante a rodada r. Ao final da *r-ésima* rodada uma árvore binária T_r é construída (BULDAS, 1998).

Todos os interessados P_i em datar um documento nesta rodada devem submeter à AD o resumo $y_{r,i}$ do documento em questão. As folhas da árvore são denotadas por diferentes $y_{r,i}$. Cada nó *K* de T_r é recursivamente denotado por $H_k = H(H_{K_L}, H_{K_R})$; onde K_L e K_R são os filhos direito e o esquerdo, respectivamente, do nó *K* e *H* é a função resumo resistente à colisão. A AD armazena apenas o *link* de rodada R_r, como mostra a Figura 2.3. A figura representa dois modos de formação de um *link* que podem ser utilizados pela AD. Todas as informações necessárias para verificar se um certo documento foi protocolado ou não durante uma certa rodada é incluso no recibo enviado para o cliente que enviou o documento para ser protocolado.

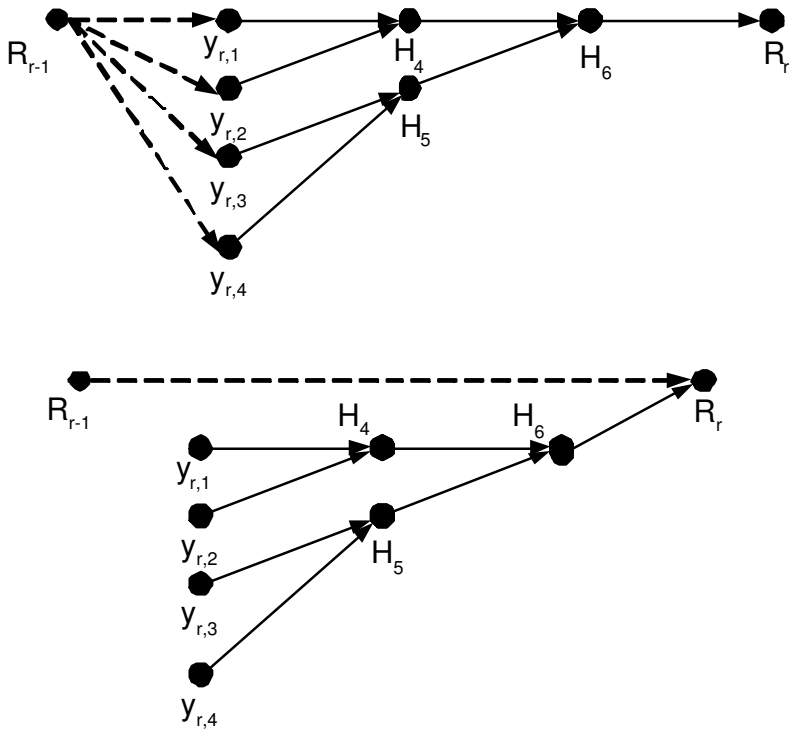

Figura 2.3: Exemplos de rodada no método da árvore.

Por exemplo, uma protocolação para $y_{r,3}$ é $[r; (y_{r,4}, L), (H_4, R)]$. O processo de verificação da protocolação $y_{r,3}$ consiste na verificação da igualdade representada na Equação 2.4.

$$R_r = H(H(H_4, H(y_{r,3}, y_{r,4})), R_{r-1}) \tag{2.4}$$

Este método de datação pode ser implementado na prática, mas possui ainda um problema, pois dois documentos devem ser datados na mesma rodada para que sua comparação temporal seja fácil e rápida, como foi proposto inicialmente.

2.4.3 Árvore Sincronizada

Para minimizar o problema do método da árvore, foi proposto um novo método de datação, chamado de Árvore Sincronizada (PASQUAL, 2002). Este método é parecido com o método explicado anteriormente, sendo que este possui alguns conceitos diferentes do método da árvore que serão explicados a seguir:

- **Rodada**: Intervalo definido pelo administrador da AD, podendo ser um intervalo de tempo ou quantidade de solicitações de protocolação recebidas. Assim, neste intervalo, solicitações de protocolação vão sendo recebidas e encadeadas para que ao final da rodada seja feita uma operação sobre os resumos dos documentos e enviado um recibo de protocolação para todos os clientes que solicitaram protocolação na mesma rodada;

- **Saltos**: No método da árvore sincronizada, os saltos são configurados como determinado período de tempo ou tamanho do arquivo do recibo, definido pelo administrador da AD. Os saltos englobam várias rodadas e podem ligar um determinado *link* ao ponto de confiança, diretamente. Como os saltos são flexíveis, pode haver sobre-saltos, ou seja, saltos maiores que englobam outros menores;

- **Ponto de Salto**[2]: É um ponto na cadeia que representa todas as rodadas anteriores ou saltos menores que estão inseridos dentro de um salto maior. Um ponto de salto é o ponto final de um salto, ou melhor, um cálculo feito com o ponto de confiança ou o último ponto de salto e a rodada anterior.

[2]Ponto de Salto: Na dissertação de Pasqual (2002), onde foi definido o método da Árvore Sincronizada, este ponto é chamado de “Ponto de Sincronismo”, mas analisando a semântica percebeu-se que melhor seria chamar este ponto de “Ponto de Salto”, já que “Ponto de Sincronismo” é mais adequado para o ponto da protocolação cruzada, ou seja, o ponto de sincronismo dos encadeamentos de diferentes ADs.

- **Recibo**: O recibo difere um pouco dos outros métodos, pois além de retornar para o cliente os resumos dos documentos de uma mesma rodada, retorna também informações de rodadas e saltos anteriores até o último ponto de confiança, podendo-se assim, a partir do recibo, reconstruir a cadeia de protocolação;

- **Ponto de confiança**: O método da árvore sincronizada define como ponto de confiança o último *link* publicado, podendo ser o resumo da cadeia originada a partir do ponto de confiança anterior. A divulgação do ponto de confiança é previamente definida pelo administrador, podendo ser divulgado, por exemplo, ao final de todos os meses. Após feita a publicação, o banco de dados da AD deve ser esvaziado, sendo os dados transportados para outra mídia qualquer, como CD ou outros discos rígidos para consultas posteriores. Para popular o banco de dados novamente, o ponto inicial será um novo ponto de confiança publicado.

Assim, os seguintes passos são seguidos para a datação de um documento no método da árvore sincronizada:

- O participante que deseja protocolar um documento calcula o resumo do mesmo e o envia para a AD;

- Quando o resumo chega à AD, esta o submete para o encadeamento junto com outros resumos recebidos em um determinado intervalo de tempo. Os resumos são encadeados através de uma função *F*, ou seja, cada resumo que chega é encadeado ao resumo H_{i-1} ou a um *link* intermediário L anterior, dependendo se o resumo que chegou é o primeiro do intervalo ou não;

- Ao final de cada rodada é aplicado novamente uma função *F* com o *link* L_r da rodada em questão e com a rodada anterior R_{r-1} ou o ponto de confiança C_j ou ainda o último ponto de salto S_n, gerando assim a rodada atual R_r.

A Figura 2.4 esquematiza o método da árvore sincronizada.

Cada H_i é o resumo de um documento encaminhado para a AD, encadeado na cadeia de datação por uma função *F*. R_r é a representação de uma rodada, ou

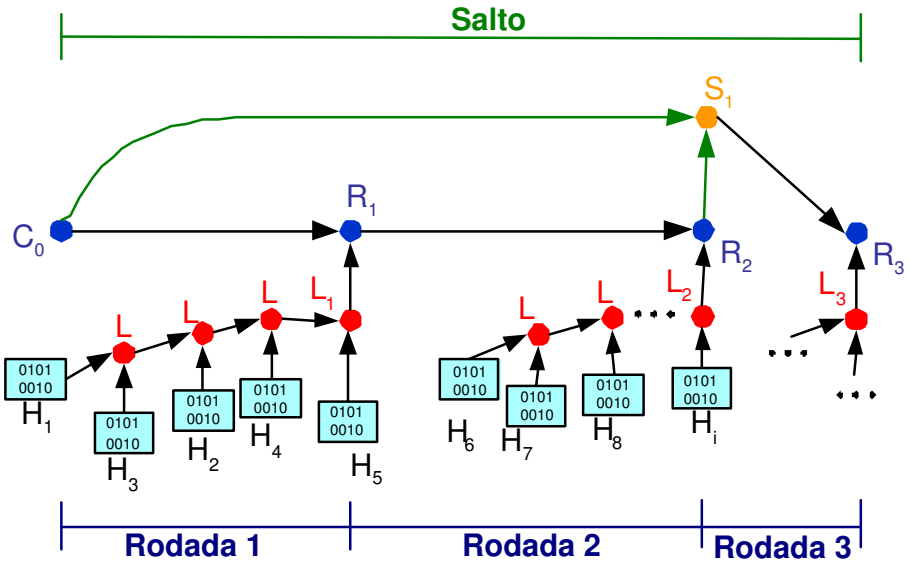

Figura 2.4: Esquema da árvore sincronizada.

seja, ele representa toda a cadeia desde o ponto de confiança até a rodada anterior e os resumos dos documentos submetidos nesta rodada. O recibo R_1 é o recibo da rodada 1, e assim por diante. O C_0 é o ponto de confiança inicial. Os *L*'s são cálculos intermediários gerados pela aplicação da função *F* nos resumos dos documentos enviados pelos clientes das ADs.

Se os saltos forem baseados no tempo, o período de tempo dos pontos de salto podem ser uma unidade como minuto, hora, dia... Estes pontos irão representar todas as rodadas contidas neste intervalo de tempo, buscando agilizar o processo de busca na árvore. Na maioria dos casos os saltos se baseiam no tamanho do recibo enviado para o cliente, ou melhor, quando o recibo chega a um determinado tamanho, um salto é criado. No recibo é armazenada uma representação da cadeia de protocolação, desde o último ponto de confiança publicado até a rodada em questão. Quanto maior a cadeia, maior será o recibo, entretanto, havendo um salto, pode-se tomar atalhos e partir da rodada em questão para o ponto de confiança ou para outro ponto de salto. Quando a quantidade de saltos chega a um determinado número, o banco de dados da AD deve ser esvaziado. A Figura 2.5 mostra um exemplo com mais detalhes dos saltos de uma árvore sincronizada. A Figura 2.6 mostra os recibos de cada rodada da árvore ilustrada na Figura 2.5.

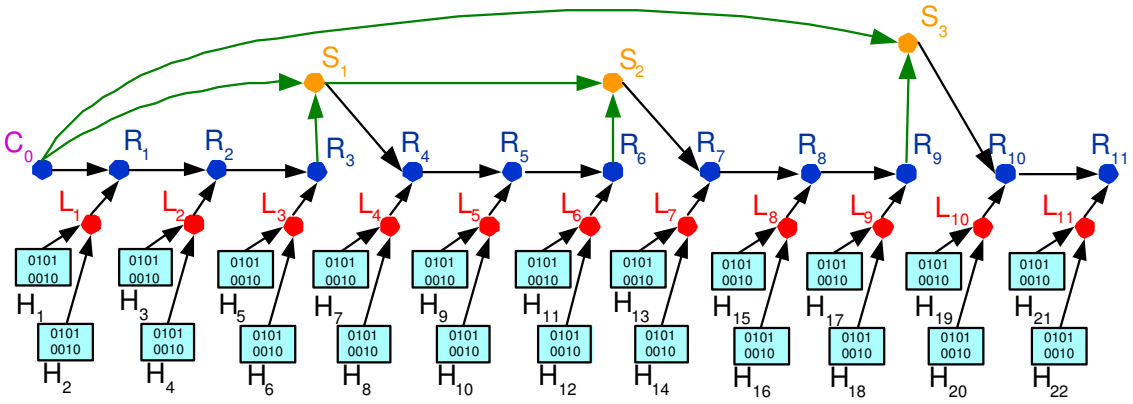

Figura 2.5: Cadeia com saltos da Árvore Sincronizada.

No caso da Figura 2.5, a cada 2 rodadas é feito um salto ou a cada dois saltos é dado um salto de nível superior.

Como mostrado na Figura 2.6, os recibos das rodadas da árvore sincronizada irão conter os resumos dos documentos que foram enviados pelos clientes naquela rodada e em caso de não haver saltos, além dos resumos dos documentos, o recibo irá conter os L_j das rodadas posteriores até o ponto de confiança, para que a cadeia possa ser reconstruída através do recibo. Caso haja saltos, o recibo irá conter os resumos dos documentos da rodada e o último R de cada salto, como mostrado nos recibos das rodadas 4 e 7 da Figura 2.6. Assim como R_1 não é armazenado no recibo da rodada 2, pois pode ser obtido a partir do C_0 e L_1, o ponto de salto S_1 também não é armazenado no recibo da rodada 4, já que pode ser obtido a partir de C_0 e R_3 que estão armazenados no recibo.

Para o caso das rodadas 4, 7 e 10, nem todos os R's precisarão ser recalculados, já que se pode pegar um "atalho" através dos saltos, os R's que estão dentro de um salto não são recalculados. Caso haja a necessidade de percorrer toda a cadeia, deve-se consultar o banco de dados da AD ou a mídia de armazenamento dos dados.

Este método atende um maior número de requisitos de segurança em relação aos métodos explanados anteriormente, segundo Pasqual (2002).

Para verificar se um documento pertence ao encadeamento, deve-se percorrer a árvore sincronizada partindo-se da localização da rodada a qual pertence o documento até o ponto de confiança que mais recentemente foi publicado. Se a partir deste

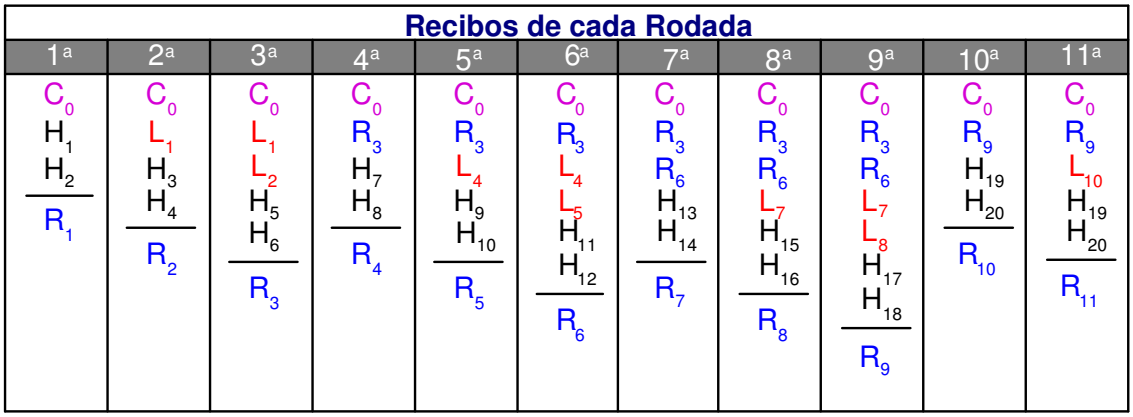

Figura 2.6: Exemplo de recibos da Árvore Sincronizada.

resumo chega-se ao ponto de confiança sem problemas, o encadeamento está íntegro. Caso a cadeia onde está o recibo não esteja mais na base de dados da AD, deve-se procurar a rodada do documento nos encadeamentos anteriores que já foram armazenados em outras mídias e realizar o mesmo procedimento descrito anteriormente. Se, a partir do resumo do documento não se chega ao ponto de confiança anterior ao documento, algum problema ocorreu na verificação ou a AD está agindo de forma maliciosa.

Além desta verificação, pode-se também ser resolvido disputas, verificando qual, entre dois ou mais documentos, foi protocolado primeiro. Este tipo de verificação pode ser muito importante em se tratando, por exemplo, de direitos autorais. O mecanismo de verificação é o mesmo descrito acima, diferenciando apenas de que nesta procura não é necessário chegar até o ponto de confiança, pois partindo de um documento e chegando ao outro, comprova-se assim qual foi protocolado primeiro.

A busca no método da Árvore Sincronizada é bem mais rápida devido aos saltos, pois não há a necessidade de percorrer todos os ramos da árvore para chegar onde se deseja, pode-se pegar atalhos através dos saltos e chegar ao ponto desejado. Se o documento desejado está dentro de um dos saltos, o algoritmo de busca deve percorrer as rodadas que foram englobadas por este salto, como é feito na busca do método do encadeamento linear. Entretanto, se há a necessidade de percorrer o interior de um salto, os mesmos problemas apresentados no método da Árvore também se aplicam, para este caso, no método da Árvore Sincronizada.

Os clientes que submeteram documentos para serem protocolados em uma mesma rodada receberão um recibo (o mesmo recibo para todos os clientes) contendo várias informações como data e hora da protocolação, informações sobre o encadeamento (como já foi explicado acima) e a assinatura digital da AD. As informações sobre o encadeamento podem ser utilizadas para que, através apenas do recibo, seja possível percorrer a árvore e chegar ao ponto de confiança, semelhante à verificação feita com o banco de dados interno da AD. A auditoria do encadeamento da AD e dos recibos de protocolação, tanto para o método da árvore sincronizada quanto para o método de encadeamento linear, faz parte da dissertação de mestrado de Costa (2003).

2.5 Conclusão

Algumas práticas realizadas no mundo real já estão sendo implementadas para o mundo virtual. Um exemplo disso é a utilização de documentos eletrônicos ao invés de documentos em papel em diversas transações.

Este capítulo trata, basicamente, das formas e métodos de datação de documentos eletrônicos mais utilizados hoje em dia.

Resumidamente, na **datação absoluta** é necessário confiar cegamente na AD e em sua fonte de tempo. Se a AD agir de forma maliciosa na datação de algum documento, é praticamente impossível de descobrir a fraude. Na forma de **datação relativa**, não há a necessidade de confiar cegamente na AD pois o resumo dos documentos protocolados são encadeados um no outro. Caso haja alguma fraude, esta poderá ser descoberta. Uma das desvantagens deste método é que o tempo absoluto não é considerado. A **forma híbrida de datação** garante tanto o tempo absoluto quanto o encadeamento dos documentos protocolados, agregando assim uma maior confiabilidade à datação.

Cabe aqui salientar o problema da validade da datação em um documento eletrônico, visto que o tempo de validade das técnicas de segurança é indeterminado. Existem técnicas que permitem que a assinatura e a protocolação digital continuem válidas mesmo que a técnica utilizada para a assinatura e ou datação já não sejam mais válidas. Estas técnicas estão descritas na dissertação de mestrado de Notoya (2002).

Capítulo 3

Protocolos de Sincronismo de Relógio

3.1 Introdução

As redes de comunicação de dados não são tão confiáveis ao ponto de transportar dados importantes sem que haja o risco de que sejam corrompidos ou atacados. A não ser que, quando necessário, seja aplicada alguma medida de segurança para que estes problemas não aconteçam ou sejam minimizados.

Algumas informações que trafegam pelas redes de comunicação de dados são de extrema importância para certas aplicações, como por exemplo a transmissão de tempo que é utilizado em transações bancárias, comércio eletrônico, bolsa de valores entre muitas outras aplicações onde o tempo preciso e íntegro é essencial.

Para que este tipo de informação trafegue pela rede com segurança, algum tipo de mecanismo que garanta a integridade dos dados deve ser aplicado. Neste contexto se encaixam os protocolos de sincronismo[1] de relógio, os quais tem a tarefa de sincronizar relógios de computadores. Cada protocolo de sincronismo possui um determinado nível de segurança dos dados e uma certa precisão.

Este capítulo descreve os protocolos de sincronismo mais conhecidos. É esclarecido na seção 3.2 como funcionam os relógios internos dos computadores. Um

[1] Sincronismo: (1) Que ocorre ao mesmo tempo, (2) Relativo aos fatos concomitantes ou contemporâneos, (3) Relativo a, ou em que há sincronia (AURÉLIO, 1999).

esclarecimento do protocolo IRIG é apresentado na seção 3.3, do protocolo ACTS na seção 3.4, do protocolo SNTP na seção 3.6, do Protocolo de Tempo na seção 3.7 e finalmente do Protocolo de Tempo do Dia na seção 3.8. A seção 3.5 apresenta apenas a citação do protocolo NTP, pois o capítulo 4 é específico para este protocolo. Este foi o protocolo escolhido para fazer o sincronismo de relógio na Infra-Estrutura de Protocolação Digital de Documentos Eletrônicos, como é detalhado no capítulo 5.

3.2 Os relógios dos computadores

As informações descritas nesta seção são baseadas em (LOMBARDI, 2002).

Desde a introdução do computador pessoal IBM-AT em 1984, todos os computadores marcam o tempo da mesma maneira, independentemente se o computador possui um processador 286, 386, 486 ou Pentium. Cada computador possui dois tipos de relógio: o relógio de *hardware* e o relógio de *software*. O relógio de *software* funciona quando o computador está ligado e permanece inativo quando o computador está desligado. O relógio de *hardware* utiliza a bateria do computador e trabalha também enquanto o computador está desligado.

O relógio de *software* é gerado por um contador de tempo *Intel 8254 timer-counter*. Este aparelho gera uma interrupção a cada 54.936 milisegundos ou em torno de 18.2 vezes por segundo. A BIOS (*Basic Input Output System*) é quem controla este relógio através de um *software* interno.

O relógio de *software* é um marcador de tempo que não pode ser confiável. Seu tempo é limitado pela instabilidade dos pedidos de interrupções. Qualquer mudança nas interrupções pode causar problemas no horário marcado pelo relógio. Mas o principal problema do relógio de *software* é quando o computador é desligado, o relógio pára e perde toda a informação que adquiriu durante o período em que estava trabalhando. Por isso o relógio de *hardware* também é necessário.

O relógio de *hardware* é baseado no chip *Motorola 146818 Real Time Clock* ou em algum equipamento equivalente. Quando o computador está desligado, o

relógio de *hardware* está trabalhando utilizando baterias.

Quando o computador é ligado ou desligado, há um sincronismo entre o relógio de *software* e o relógio de *hardware* ou vice-versa. Neste sincronismo, há a perda de alguns milisegundos. O relógio de *software* conta valores no intervalo de 55 milisegundos, já o relógio de *hardware* conta valores de 1 segundo de diferença, não pode mostrar frações de segundo. Esta característica do relógio de *hardware* é determinada pela qualidade do cristal oscilador. Além disso, alguns fatores externos podem influenciar na oscilação do cristal, o que faz com que o relógio perca a precisão em cerca de 1 segundo por dia. Geralmente o relógio de *software* se baseia no relógio de *hardware* para trabalhar, o que faz com que também perca ainda mais precisão.

Com isso, para transações que necessitam de um tempo mais preciso é necessário que os computadores se sincronizem com fontes de tempo seguras através de protocolos de sincronismo para que o tempo utilizado nas transações eletrônicas também seja confiável.

Para este tipo de necessidade, vários protocolos de transmissão de tempo foram criados. A disseminação de informação de tempo pode ser realizada através de sinais de rádio, por linha telefônica ou através da Internet. Existem muitos protocolos de distribuição de tempo, conforme detalhado no apêndice do relatório técnico da empresa *Hewlett-Packard* (HP) (PACKARD, 2003). As seções seguintes irão descrever alguns destes protocolos.

3.3 IRIG

O IRIG (*Inter-Range Instrumentation Group*) é um protocolo que transmite informações temporais através de sinais de rádio. Este padrão define as características de códigos seriais de tempo que são utilizadas pelas agências governamentais dos Estados Unidos e indústrias privadas.

Este protocolo é formado por 3 códigos ou palavras. A primeira palavra é o “tempo do ano” (*time-of-year*) na notação BCD (*Binary Coded Decimal*): dias, horas, minutos, segundos e frações de segundo, dependendo da estrutura utilizada. A segunda

palavra é reservada para codificação de controles, especificações entre outros. A terceira palavra são os “segundos do dia” (*seconds-of-day*) na notação SBS (*Straight Binary Seconds*). A maioria dos aparelhos atualmente só utilizam a primeira palavra (IRIG, 1987).

O NIST (*National Institute of Standards and Technology* - Instituto Nacional de Padrões e Tecnologia) possui outras formas de distribuição de tempo via rádio como o WWV, o WWVH e o WWVB, todos são estações de rádio do NIST que transmitem o horário todos os dias, 24hs por dia (NIST, 2003).

3.4 Serviço Automatizado de Tempo em Computador

O protocolo de Serviço Automatizado de Tempo em Computador (*Automated Computer Time Service* - ACTS) permite o sincronismo de relógios através de linha telefônica e modem analógico. O computador que deseja sincronizar seu relógio, ao conectar-se com um servidor de tempo através do ACTS, recebe a informação temporal em código ASCII (*American Standard Code for Information Interchange* - Código Americano Padrão para Troca de Mensagens). A informação do tempo contida neste código é utilizada para atualizar o relógio do computador (LOMBARDI, 2002). Este protocolo foi criado pelo NIST.

O último caracter no código recebido é o asterisco (*). O asterisco é chamado de marcador do tempo (*on-time marker* - OTM). O valor do tempo enviado refere-se ao tempo quando o OTM chega no computador requisitante, ou melhor, o tempo recebido é o tempo em que a solicitação chega ao servidor ACTS acrescido de alguns ajustes como o tempo de tráfego e o tempo de enviar a resposta ao computador requisitante. Para o cálculo destes tempos há uma troca de OTMs entre o ACTS e o computador requisitante.

O formato do tempo enviado pelo ACTS é o seguinte (LOMBARDI, 2002):

JJJJJ YR-MO-DA HH:MM:SS TT L UT1 msADV UTC(NIST) <OTM>

Onde:

JJJJJ são os últimos 5 dígitos da data Juliana[2];

YR-MO-DA é a data em ano, mês e dia;

HH:MM:SS são as horas, minutos e segundos, respectivamente, no Tempo Universal Coordenado (UTC);

TT são dois dígitos que indicam se os Estados Unidos está em horário de verão ou não;

L é um dígito que indica se há a necessidade de acrescentar o *leap second*[3];

UT1 é um fator para corrigir o antigo UTC para o atual;

msADV é um código de 5 dígitos que indica o número de milisegundos que o NIST acrescentou no tempo antes de enviá-lo;

UTC(NIST) é um rótulo que todo código de tempo possui;

<**OTM**> é o caracter enviado no final do código do tempo.

O WWV e o WWVH, além de transmitirem o tempo via rádio, podem ser ouvidos por telefone.

3.5 Protocolo de Tempo em Rede

O Protocolo de Tempo em Rede (*Network Time Protocol*- NTP) é um dos mais importantes e utilizados protocolos de sincronismo de tempo via Internet.

[2] A Data Juliana simplesmente conta os dias desde 1^o de janeiro de 4713 antes de Cristo até os dias atuais.

[3] *Leap second* (segundo intercalado): Diferença entre o horário que utilizamos atualmente (UTC) - baseado no relógio atômico - e as atividades humanas - baseadas na rotação de nosso planeta, a qual possui uma fração de segundo mais rápida ou mais lenta. Assim, com o acúmulo desta diferença, tornou-se necessário correções de tal forma que o tempo transmitido (UTC) não se afaste muito do Tempo Universal (tempo da terra) (ON, 2003).

Devido às suas vantagens perante os outros protocolos, que serão explanadas posteriormente, o NTP foi o escolhido para fazer o sincronismo do relógio dos elementos pertencentes à Infra-Estrutura de Protocolação Digital de Documentos Eletrônicos. Assim, o capítulo 4 destina-se exclusivamente para este protocolo, descrevendo suas características, funcionalidades e a razão pela qual este foi escolhido para fazer parte da infra-estrutura proposta neste trabalho.

3.6 Protocolo de Tempo em Rede Simples

O Protocolo de Tempo em Rede Simples (*Simple Network Time Protocol* - SNTP) é uma adaptação do NTP. Ambos são utilizados para sincronizar relógios de computadores via Internet. O SNTP é utilizado quando o NTP completo não é necessário. Este protocolo não modifica as características do NTP, apenas algumas configurações são mudadas para que este protocolo trabalhe de maneira mais simples. O SNTP é um protocolo mais simples que o NTP para a sincronização de clientes e servidores (MILLS, 1995).

É recomendado que o SNTP seja utilizado apenas em extremidades de redes de sincronismo, os clientes devem operar apenas em níveis de *stratum*[4] altos, nenhum cliente SNTP deve depender de outro cliente para o sincronismo de seu relógio. Já os servidores SNTP devem trabalhar de forma que seu *stratum* seja muito baixo, onde alguma outra fonte de tempo confiável esteja disponível.

Como o NTP, o SNTP pode operar nos modos *unicast* (ponto a ponto) e *broadcast* (de um ponto para todos os pontos da rede). Um cliente ponto a ponto envia uma requisição para o servidor e espera uma resposta que determina o tempo e, opcionalmente, o atraso e a compensação do relógio local relativa ao servidor. Já o servidor *broadcast* envia uma mensagem periodicamente para o grupo de endereço IP[5] *broadcast*

[4] *Stratum*: Distância entre a fonte de tempo confiável e um computador pertencente a rede de sincronismo NTP, quanto mais baixo o *stratum* mais perto da fonte de tempo está o servidor. Será melhor explicado no capítulo 4.

[5] IP: Protocolo da Internet - *Internet Protocol*.

ou *multicast* (de um ponto para alguns pontos da rede) e não espera pedidos dos clientes, os clientes ficam esperando estas mensagens que são enviadas pelo servidor.

O SNTP utiliza o padrão do NTP como formato das datações (*timestamp*). As datações do NTP são representadas como um número de ponteiro fixo sem sinal de 64 bits, representando os segundos relativos a 0h do dia primeiro de janeiro de 1900. A precisão desta notação é de 200 picosegundos (MILLS, 1995). O formato das mensagens do SNTP são idênticas ao formato das mensagens do NTP, a única diferença é que alguns campos são inicializados com valores pré definidos. Os campos das mensagens NTP poderão ser vistos com detalhes na Figura 4.1 do capítulo 4, página 41.

3.7 *Protocolo de Tempo*

O Protocolo de Tempo (*Time Protocol*) transmite o tempo através da Internet. Pode ser utilizado sobre o Protocolo de Controle de Transmissão (*Transmission Control Protocol* - TCP) ou o Protocolo de Pacote do Usuário (*User Datagram Protocol* - UDP). Quando utilizado via TCP, o servidor fica “ouvindo” a porta 37. Quando uma conexão é estabelecida, o servidor retorna um valor de 32 bits e fecha a conexão. De maneira análoga acontece com o sincronismo via UDP, o servidor espera por um pacote na porta 37 e quando a conexão é estabelecida, retorna um pacote de 32 bits. Em ambos os casos, se o servidor não está apto para fornecer o tempo ele recusa a conexão.

O tempo é o número de segundos desde a meia noite do dia primeiro de janeiro de 1900 (no padrão GMT). Esta base poderá servir até o ano de 2036. Por exemplo, o tempo 2.629.584.000 corresponde o horário de 00:00hs do dia primeiro de maio de 1983 GMT (POSTEL; HARRENSTIEN, 1983).

3.8 *Protocolo de Tempo do Dia*

O Protocolo de Tempo do Dia (*Daytime protocol*) envia a data e tempo através da Internet como uma *string* de caracteres (POSTEL, 1983).

A conexão pode ser via TCP ou UDP. Via TCP, o servidor espera por

uma conexão na porta 13. Quando esta conexão acontecer, a data e a hora são enviadas como uma *string* de caracteres ASCII. A conexão é encerrada quando o tempo é transmitido. Nas conexões via UDP, o servidor fica esperando os pacotes UDP na porta 13. Quando um pacote chega, o servidor envia a data e a hora corrente na forma de uma *string* de caracteres ASCII, como acontece com a conexão TCP (POSTEL, 1983).

Não há uma sintaxe específica para o Protocolo de Tempo do Dia. As sintaxes comumente utilizadas são:

- Dia da semana, mês dia, ano horário - zona de tempo. Por exemplo: *Tuesday, February 22, 1982 17:37:43-PST*;
- dd mmm yy hh:mm:ss zzz. Por exemplo: *02 FEB 82 07:59:01 PST*.

3.9 Conclusão

O horário do mundo real precisa estar presente também no mundo virtual, visto que várias transações são realizadas através de computadores.

Este capítulo se preocupa em mostrar os protocolos responsáveis por sincronizar o relógio dos computadores com uma ou mais fontes de tempo confiáveis.

O capítulo trata dos protocolos que possuem código fonte aberto e que são mais utilizados atualmente. Deu-se maior ênfase aos protocolos que utilizam a Internet como meio de transmissão de dados, por serem mais relevantes ao presente trabalho.

Capítulo 4

Protocolo de Tempo em Rede

4.1 Introdução

Como visto no capítulo 3, existem muitos protocolos que fazem a transmissão da hora oficial de um país para o mundo virtual. O tempo pode ser transmitido através de linha telefônica, via ondas de rádio e via Internet.

Para a transmissão do tempo através de linha telefônica é necessário um modem e uma linha telefônica. Para que a transmissão aconteça, deve-se fazer uma ligação do computador de origem para uma fonte de tempo que disponibilize o acesso através desta via de transmissão. Para o sincronismo de relógios no contexto deste trabalho, este protocolo talvez não seja o mais indicado, por agregar um alto custo ao sincronismo, já que uma ligação telefônica é feita para o local onde se encontra o servidor destino.

Os protocolos baseados em transmissão via ondas de rádio necessitam de um equipamento receptor de rádio freqüência e antenas. Estes equipamentos apresentam um elevado custo, o que torna a utilização dos protocolos que utilizam transmissão do tempo via rádio impraticáveis para a maioria das instituições.

Assim, uma das formas mais interessantes de se sincronizar os relógios dos computadores é através da Internet, devido ao reduzido custo econômico e aos inúmeros e crescentes estudos relacionados a esta área.

O protocolo de sincronismo de relógios através da Internet mais utili-

zado é o Protocolo de Tempo em Rede (*Network Time Protocol* - NTP). Este possui várias características que o tornam mais completo que os outros protocolos; possui mais recursos para a autenticação, segurança, alta precisão, entre outras vantagens perante os outros protocolos de sincronismo pela Internet. Devido a estas características, que são apresentadas ao longo deste capítulo, este protocolo foi escolhido para fazer o sincronismo dos relógios dos elementos pertencentes à infra-estrutura proposta neste trabalho.

Na seção 4.2 dá-se uma introdução ao NTP, seu comportamento, características e como trabalha. A hierarquia do NTP é apresentada na seção 4.3. Para que o relógio de um computador seja sincronizado através do NTP, este calcula vários fatores necessários para o sincronismo antes que o relógio do computador seja alterado. A descrição dos cálculos realizados e como é feito o ajuste do relógio estão descritos na seção 4.4. Os dois pontos que realmente são significantes a este trabalho são os aspectos de segurança do NTP - descritos na seção 4.5 - e o protocolo *Autokey* que é responsável pela autenticação dos servidores de tempo e das mensagens trocadas - descrito na seção 4.6. Ao final deste capítulo, na seção 4.7, tem-se a conclusão do mesmo.

4.2 Introdução ao NTP

O Protocolo de Tempo em Rede (*Network Time Protocol* - NTP) é utilizado para sincronizar relógios de computadores através da Internet. Ele possui mecanismos para acessar os serviços de tempo disponibilizados por uma fonte de tempo.

O protocolo NTP teve sua primeira versão desenvolvida na Universidade de Maryland em 1985 por Louis Mamakos e Michel Petry. A primeira especificação formal foi apresentada na RFC[1]958. Atualmente, está sendo implementada a versão 4 deste protocolo, projeto coordenado por David L. Mills na Universidade de Delaware, Estados Unidos. A versão 4 do protocolo ainda não virou uma RFC, mas sua documentação e implementação podem ser encontradas em (MILLS, 2003e). A arquitetura, protocolos e algoritmos da versão 3 do NTP são especificados na RFC1305 (MILLS, 1992). As versões

[1]RFC: *Request For Comments* são padrões que definem a Internet e como ela opera. RFC também se refere à maneira como estes documentos são discutidos e aprovados pela comunidade da Internet.

mais novas do NTP são compatíveis com as versões anteriores.

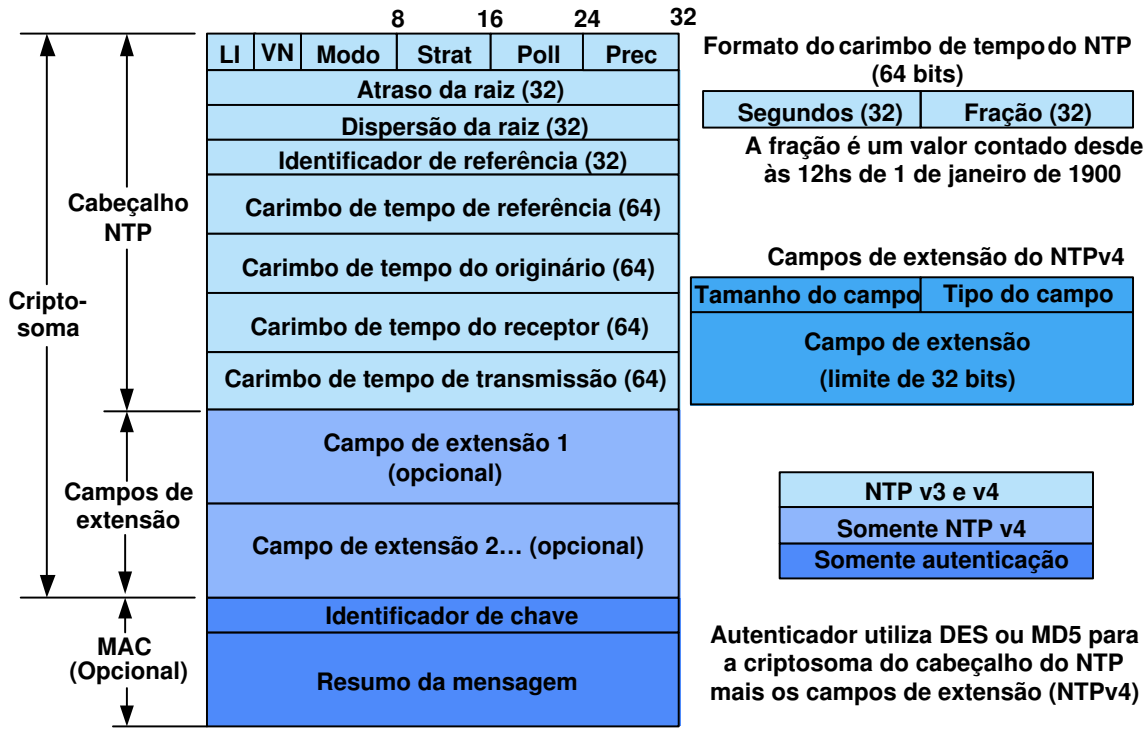

Figura 4.1: Formato das mensagens NTP.

O NTP possui uma precisão de milisegundos em LANs[2] se estiverem em boas condições, ou seja, sem muitos roteadores ou fontes de atraso, e em torno de 10-100 milisegundos em WANs[3]. Essa precisão pode diminuir caso a rede possua muitos *hubs*, *switches*, roteadores ou muito tráfego na rede (DEETHS; BRUNETTE, 2001a). O NTP possui suporte tanto para criptografia simétrica quanto para criptografia assimétrica, o que previne acidentes ou ataques maliciosos. Requer pouco recurso computacional, ou seja, os servidores NTP podem servir a vários clientes utilizando muito pouco de sua capacidade computacional. A largura de banda exigida é mínima.

A Figura 4.1 representa o formato das mensagens trocadas entre servidor e cliente de tempo NTP. Para a requisição de tempo, alguns campos do cabeçalho vão do cliente para o servidor vazios, para que o servidor preencha estes campos com os dados

[2]LAN: Rede de área local - *Local Area Network*.

[3]WAN: Rede de grande área - *Wide Area Network*.

relevantes à solicitação do cliente, incluindo o seu tempo (MILLS, 1992). Os campos da requisição de tempo são enviados vazios para que o tamanho dos pacotes da requisição e da resposta seja o mesmo ou bem parecidos.

O significado de cada campo do cabeçalho da mensagem está detalhado na Tabela 4.1, retirado da RFC1305 (MILLS, 1992).

Tabela 4.1: Campos do cabeçalho das mensagens NTP

Campos	**N^o de bits**	**Descrição**
LI: Indicador de Segundo Intercalado (*Leap Indicator*)	2	Indica o ajuste feito no final do dia (00: nenhum, 01: adiciona um segundo, 10: diminui um segundo, 11: relógio não sincronizado).
VN: Número da versão (*Version Number*)	3	Número da versão (a versão atual é a 4).
Mode: Modo	3	1 - Simétrico Ativo (que inicia a conexão); 2 - Simétrico Passivo; 3 - Cliente; 4 - Servidor; 5 - *Broadcast* (servidor informa ao cliente que o servidor pode prover o tempo); 6 - Mensagens de controle NTP; 7 - Reservado para uso privado.
Strat: *stratum*	8	Indica o *stratum* que o servidor pertence. 0 - Não especificado; 1 - Referência primária (como relógio atômico, por exemplo); 2..15 - Referência secundária (via NTP).
Poll: apuração	8	Indica o intervalo máximo entre mensagens sucessivas (expressado como uma potência de dois).
Prec: Precisão (*Precision*)	8	Precisão do relógio local em segundos (expressado como uma potência de 2: -10 significa 2^{-10} ou 1/1024 = 0.97ms).
Root delay: atraso da raiz	32	Indica o tempo de atraso de uma requisição para a referência de tempo primária e resposta enviada por ele, em segundos. São permitidos valores positivos ou negativos, dependendo da precisão do relógio.
Root Dispersion: Dispersão da raiz	32	Erro máximo relativo à fonte primária de tempo, em segundos. Apenas valores positivos são permitidos.
Reference Identifier: Identificador de referência	32	Identificador da referência de tempo: para um servidor primário utiliza-se um código ASCII de no máximo 4 octetos; para outros servidores este campo é preenchido com os 4 octetos referentes ao endereço IP do servidor.
Reference Timestamp: Carimbo de tempo de referência	64	Tempo local em que o relógio foi ajustado pela última vez.

Continua na próxima página...

Tabela 4.1 – ***Continuação da página anterior***

Campos	Nº de bits	Descrição
Originate Timestamp: Carimbo de tempo na origem	64	Tempo de partida da requisição do cliente para o servidor de tempo.
Receive Timestamp: Carimbo de tempo na recepção	64	Tempo de chegada da requisição de tempo no servidor.
Transmit Timestamp: Carimbo de tempo na transmissão	64	Tempo de saída da resposta do servidor para a requisição de tempo do cliente.

Os carimbos de tempo são determinados copiando o valor corrente do relógio local quando algum evento significante acontece, como por exemplo a chegada de uma mensagem. O carimbo de tempo pode ser refeito a cada "nó" (roteadores, servidores, ...) da transmissão. Se algum campo de carimbo de tempo não é preenchido, pelo tempo não estar disponível no computador, por exemplo, no campo de 64 bits é marcado zero, indicando que o valor é inválido ou indefinido (MILLS, 1992). Pacotes que não estão protegidos não possuem o campo de autenticação. Pacotes que trafegam protegidos com a criptografia simétrica utilizam o algoritmo MD5 como função resumo para calcular o MAC (Código de Autenticação de Mensagem - *Message Authentication Code*) - que é a função resumo dos campos de extensão e do cabeçalho NTP - e é encaminhado anexado à mensagem. Os pacotes protegidos com criptografia assimétrica, da mesma forma como acontece com a criptografia simétrica, incluem o MAC e os campos de extensão (MILLS, 2003d).

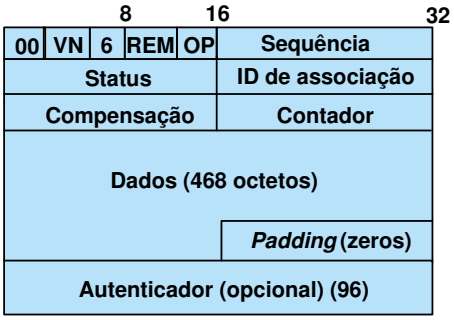

Figura 4.2: Cabeçalho das mensagens de controle NTP.

Mensagens de controle representam uma opção de gerenciamento da rede NTP, caso não haja algum outro protocolo que desempenhe esse papel tal como o

SNMP (Protocolo Simples de Gerenciamento de Rede - *Simple Network Management Protocol*), por exemplo. Com a mensagem de controle pode-se fazer rotinas de controle e monitoramento, ajustar parâmetros e monitorar operações regulares. A mensagem de controle NTP tem o valor 6 especificado no campo “Modo”, no primeiro octeto do cabeçalho. É formatado conforme a Figura 4.2.

A Tabela 4.2 mapeia o significado de cada campo do cabeçalho das mensagens de controle NTP.

Tabela 4.2: Campos do cabeçalho das mensagens de controle NTP

Campos	**N^o de bits**	**Descrição**
VN: Número da versão (*Version Number*)	3	A versão atual é a 4.
Mode: Modo	3	Indica o modo. Neste caso ele sempre tem o valor 6, indicando que é uma mensagem de controle.
Response Bit(R): Bit de resposta	1	Preenchido com zero para comando e com um para resposta.
Error Bit(E): Bit de erro	1	Preenchido com zero para resposta normal e com um para erro.
More Bit(E): Bit de mais	1	Preenchido com zero para último fragmento e com um para todos os outros.
Operation Code(Op): Código de operação	5	0 - Reservado; 1 - Estado lido para comandos/respostas; 2 - Variáveis lidas para comandos/respostas; 3 - Variáveis escritas comandos/respostas; 4 - Variáveis do relógio lidas para comandos/respostas; 5 - Variáveis do relógio escrita para comandos/respostas; 6 - Configuração da porta/endereço de bloqueio comando/resposta; 7 - Resposta de bloqueio; 8..31 - Reservado.
Sequence: Seqüência	16	Indica o número de seqüência de comandos ou respostas.
Status	16	Indica o estado corrente do sistema, par[4] ou servidor com quem se está fazendo o sincronismo ou do relógio, com valores codificados.
Association ID: ID de associação	16	Indica uma associação válida.
Offset: Compensação	16	Indica a compensação, em octetos, do primeiro octeto na área de dados.
Count: Contador	16	Indica o tamanho do campo de dados em octetos.

Continua na próxima página...

[4]Par: Um computador que deseja sincronizar seu relógio com o relógio de outro computador pertencente ao mesmo *stratum*, não havendo cliente e servidor previamente definidos.

Tabela 4.2 – *Continuação da página anterior*

Campos	Nº de bits	Descrição
Data: Dados	≤468 octetos	Contém o dado da mensagem para o comando ou resposta.
Authenticator: Autenticador (opcional)		Quando o mecanismo de autenticação do NTP é implementado, este campo contém a informação de autenticação.

O NTP pressupõe que pelo menos uma fonte de tempo seja confiável dentre as várias que um servidor NTP pode possuir. O tempo oferecido por uma fonte de tempo confiável deve ser transmitido para toda a hierarquia NTP de forma segura e precisa. Múltiplos servidores e pares fornecem redundância e diversidade.

Seu princípio é que um cliente interessado em sincronizar o seu relógio obtenha o tempo de diversos servidores de tempo, através de diferentes caminhos de rede, fazendo com que os erros de transmissão sejam minimizados. A garantia de serviço do protocolo NTP é, justamente, a redundância de servidores e a inserção de carimbos de tempo nas mensagens, o que torna o NTP resistente a falhas e ataques de repetição de mensagens (DIAS; CUSTÓDIO; DEMÉTRIO, 2003).

O protocolo pode operar nos seguintes modos:

- **Cliente/Servidor**: O cliente requisita informações de sincronismo para servidor que está mais próximo de uma fonte de tempo;

- **Ponto a Ponto**: Uma mesma entidade pode atuar tanto como cliente quanto servidor de tempo;

- ***Multicast/Broadcast***: Permite a descoberta de servidores NTP através de mensagens enviadas de modo *multicast* ou *broadcast* para os computadores que desejam sincronizar seu relógio. O modo *multicast* envia mensagens de um ponto para alguns pontos específicos da rede, já o modo *broadcast* envia mensagens de um ponto para todos os pontos da rede, ou melhor, deixa a mensagem “vagando” na rede disponível para qualquer elemento pertencente à mesma.

No topo da hierarquia NTP existem um ou mais relógios de referência. Relógios de referência são assumidos como precisos (DEETHS; BRUNETTE, 2001a).

4.3 Hierarquia NTP

O NTP utiliza o protocolo UDP na porta 123 para comunicação entre clientes e servidores. São feitas tentativas de comunicação em intervalos pré-estabelecidos até que o servidor ou o par responda (DEETHS; BRUNETTE, 2001a).

O NTP trabalha com um modelo de hierarquia onde um pequeno número de servidores disponibiliza seu tempo a um grande número de clientes. Os clientes em cada nível ou *stratum* são potenciais servidores para outros clientes de outra *strata*. O conjunto de *stratum* é chamado de *strata*. O número do *stratum* aumenta do primeiro servidor (*stratum* 1) que está conectado à fonte de tempo confiável até o décimo quinto (*stratum* 15), considerando-se que o *stratum* zero é a fonte de tempo confiável (um relógio atômico, por exemplo) e que a partir do décimo quinto *stratum* é considerado infinito; na prática é muito difícil encontrar clientes com este número de *stratum*. Assim, uma árvore vai sendo montada, sendo que os nós desta árvore são os servidores de tempo e as folhas são os clientes finais que não fornecem seu tempo a nenhum outro computador. Os clientes podem determinar automaticamente, através do NTP, a melhor fonte de tempo entre os múltiplos servidores que possuem e prevenir más fontes de tempo, o que será melhor explicado posteriormente. A Figura 4.3 mostra a hierarquia do NTP.

Os servidores conectados à fonte de tempo (*stratum* 0) possuem *stratum* 1. Entretanto, alguns relógios comerciais possuem interfaces para conexões externas, fazendo com que estes relógios atuem como *stratum* 1. Um servidor/cliente de tempo conectado a um servidor do *stratum* 1 pertence ao *stratum* 2 e assim por diante.

É muito importante utilizar vários servidores pois nada impede que um servidor de tempo utilize como referência o seu relógio interno ao invés de utilizar o horário proveniente de uma fonte confiável.

De acordo com o relatório da empresa Sun (DEETHS; BRUNETTE, 2001a), mais da metade dos clientes conectados na Internet que utilizam NTP pertencem ao *stra-*

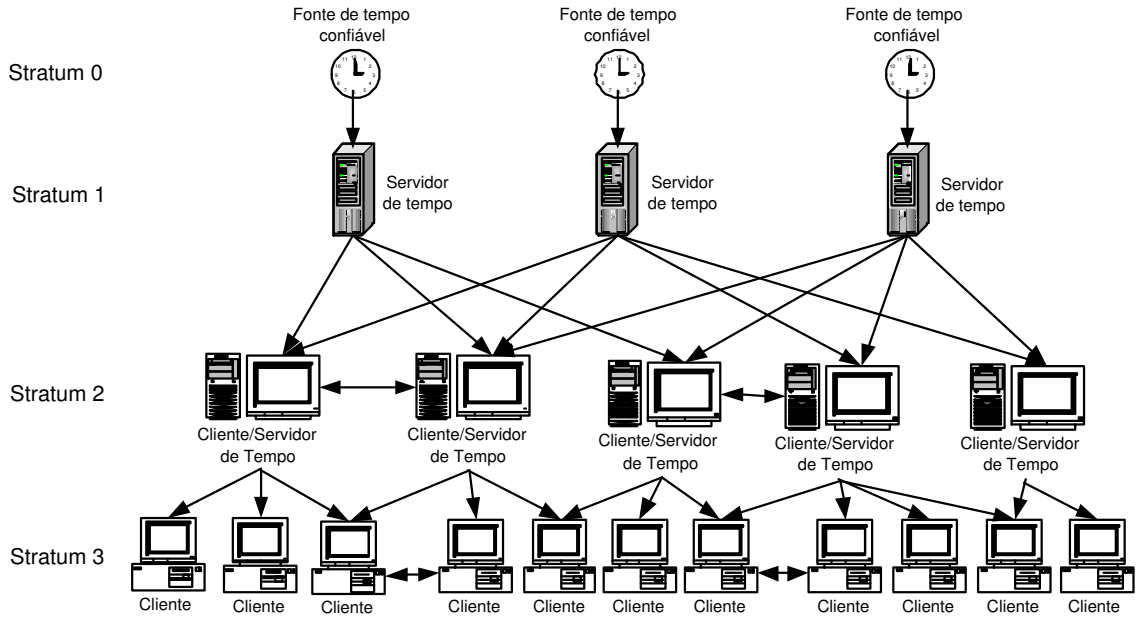

Figura 4.3: Hierarquia típica do NTP.

tum 3, o restante pertence a *strata* 2 e 4. Há um número baixo de servidores de *stratum* 1, devido ao seu custo e à sua manutenção. Assim, os servidores de *stratum* 2 são mantidos como públicos e podem ser disponibilizados como serviço público.

4.4 Ajuste do relógio

Antes do ajuste do relógio, o NTP faz a seleção do melhor tempo dentre os servidores disponíveis. Muitos passos são envolvidos para determinar o tempo correto.

São executados vários testes para garantir que o tempo escolhido será o correto. Abaixo estão listadas, em ordem, as verificações realizadas (DEETHS; BRUNETTE, 2001b).

1. **Verificações de Sanidade**: Garantem que os pacotes são válidos;

2. **Filtragem**: Usa o histórico de exemplos para um dado relógio para reduzir a oscilação do mesmo;

3. **Algoritmos de Intersecção**: Remove os relógios que estão configurados com o

tempo errado;

4. **Verificação dos candidatos para agrupamento**: Remove os piores relógios até sobrarem os 10 melhores (para limitar o processamento);

5. **Sincronismo da fonte selecionada do algoritmo de agrupamento**: Seleciona a melhor fonte dos 10 restantes;

6. **Combinação do relógio**: Revisa o tempo da melhor fonte baseado no tempo e estimativas de erro de outros relógios.

O NTP garante que os pacotes recebidos pelos clientes e servidores são válidos antes de seu processamento, através das verificações de sanidade. Pressupõe-se que qualquer servidor que sobrevive às verificações de sanidade está configurado corretamente. As seguintes verificações de sanidade são executadas para cada pacote NTP pelo cliente (DEETHS; BRUNETTE, 2001b):

- **Pacotes do mesmo servidor não podem ser duplicados**: Este teste elimina a possibilidade de um pacote chegar através de dois roteadores e ser processado duas vezes;

- **A seqüência lógica dos carimbos de tempo do cliente deve estar correta**: Esta é uma verificação de consistência de que o servidor está respondendo a última requisição do cliente. Isto evita que o servidor receba pacotes fora de ordem. Pode ser também uma proteção contra pacotes forjados;

- **O servidor é alcançável**: Os servidores não alcançáveis não são eliminados da lista de servidores até as verificações de sanidade;

- **O cálculo do atraso da rodada e a dispersão para o servidor precisa ser menor que 16 segundos**: Se o erro de abrangência do relógio (primeiramente determinado pela latência da rede) é maior que 16 segundos, então o NTP considera o relógio inadequado para o sincronismo;

- **A autenticação precisa ser feita com sucesso (se configurado)**: Se o cliente é apenas configurado para aceitar pacotes autenticados, pacotes não autenticados são negados durante as verificações de sanidade;

- **O relógio do servidor precisa ser sincronizado com uma fonte externa que tenha sido atualizada no último dia**: NTP irá ignorar pacotes de servidores que não estão sincronizados com o *stratum* 0. A restrição de um dia permite o NTP perder conectividade de 24 horas a ainda prover serviços para clientes NTP;

- **O servidor precisa ser de *stratum* mais baixo ou igual ao do cliente**: Isto previne ciclos na configuração NTP;

- **O atraso total e o erro máximo precisa ser menos que 16 segundos do relógio raiz**: Isto previne configurações NTP imprecisas onde existem grandes latências em vários níveis da hierarquia.

Uma vez os pacotes passados pelas verificações de sanidade, alguns processamentos de alto nível são executados.

Um cliente NTP pode se relacionar com 64 servidores, mas apenas 10 destes é que são potenciais fontes de tempo. O NTP não apenas tenta sincronizar o tempo correto, ele também calcula uma variedade de erros no tempo dos clientes. De fato, esta variedade de erros é amarrada à escolha do cliente pelo servidor de tempo. O erro máximo em qualquer direção é chamado de dispersão. O NTP determina o melhor tempo baseado em muitos fatores como a compensação (*offset*), o atraso (*delay*) e um fator de erro (DEETHS; BRUNETTE, 2001b).

A compensação é metade da diferença do intervalo entre a requisição de tempo feita pelo cliente e a resposta do servidor. Esta diferença é medida no momento em que o pacote chega e deixa o servidor e quando o pacote deixa e chega no cliente. O atraso é o tempo para o cliente receber a resposta do servidor relativa à requisição de tempo enviada anteriormente pelo cliente. O tempo gasto com o processamento feito pelo servidor é subtraído do atraso, pois o atraso trata apenas a questão da rede. O fator de erro representa erros encontrados na leitura do relógio e tolerância de freqüência. A

dispersão (*dispersion*) pode ser representada como metade do atraso mais o fator de erro. A dispersão representa o máximo erro possível de compensação. O tempo corrente do cliente é representado com compensação zero. A compensação representa o ajuste no tempo corrente do cliente, ou seja, é a mudança necessária no relógio do cliente de acordo com os cálculos realizados para determinar o servidor mais adequado para o sincronismo (DEETHS; BRUNETTE, 2001b). A Equação 4.1 mostra como os fatores acima explanados relacionam-se entre si.

$$c - (\frac{a}{2} + e) \leq aj \leq c + (\frac{a}{2} + e) \tag{4.1}$$

Onde:

c = compensação;

a = atraso;

e = erro;

aj = ajuste do tempo atual.

As verificações de sanidade são realizadas por filtros e algoritmos internos de cada cliente. Quando um pacote de tempo é recebido pelo cliente, o pacote fica aguardando em uma fila de pacotes da mesma fonte. A fila possui 8 posições e quando um novo pacote chega, o mais velho é descartado. Os algoritmos de filtro são utilizados para diminuir o efeito de pequenos erros na precisão do relógio. A saída do algoritmo de filtro são valores que representam a melhor suposição da compensação corrente e o erro máximo de um determinado relógio (DEETHS; BRUNETTE, 2001b).

Os **algoritmos de filtro** selecionam o melhor tempo dentre 8 amostras de tempo obtidas do mesmo servidor de tempo, de acordo com amostras de compensação e atraso, relacionados ao *stratum* do par ou servidor, ao atraso de rede dentre outros. **O algoritmo de seleção** descarta os tempos que aparentemente estão errados. **O algoritmo de aglomeração** descarta os piores servidores mantendo no máximo os 10 melhores servidores de tempo. O **algoritmo de combinação** seleciona o melhor tempo e faz a correlação entre o relógio do cliente e todos os servidores que passaram pelo algoritmo de aglomeração. O **filtro de *loop*** e o **Oscilador de Freqüência Variável** (*Variable Frequency*

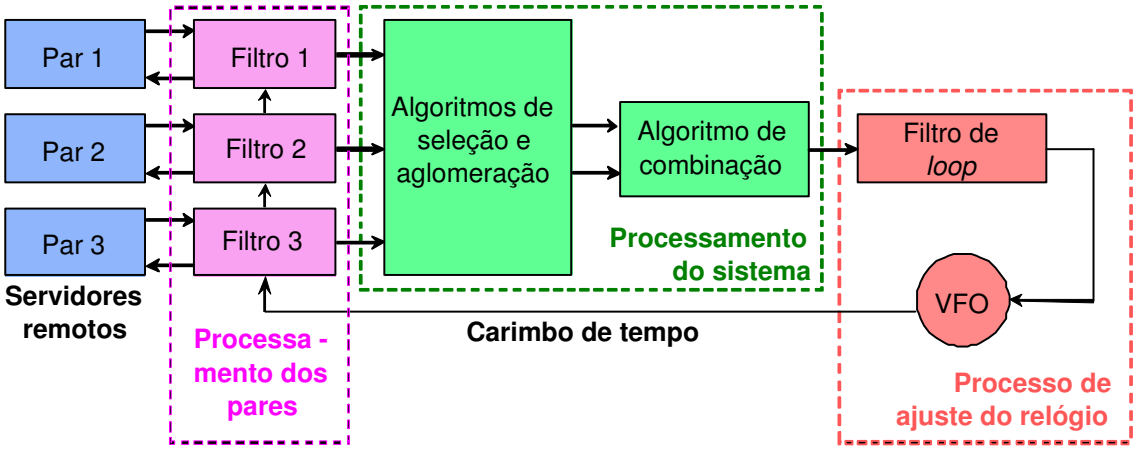

Figura 4.4: Processamento do tempo.

Oscillator - VFO) implementam um laço de reação híbrida de fase/trava-de-freqüência que determinam o tempo do sistema e também minimizam os erros e oscilações, fazendo com que o relógio do sistema mantenha seu tempo sempre próximo ao tempo escolhido anteriormente (MILLS, 2003b; DEETHS; BRUNETTE, 2001b; MILLS, 2003d, 2003c). A Figura 4.4 mostra como a filtragem do tempo acontece.

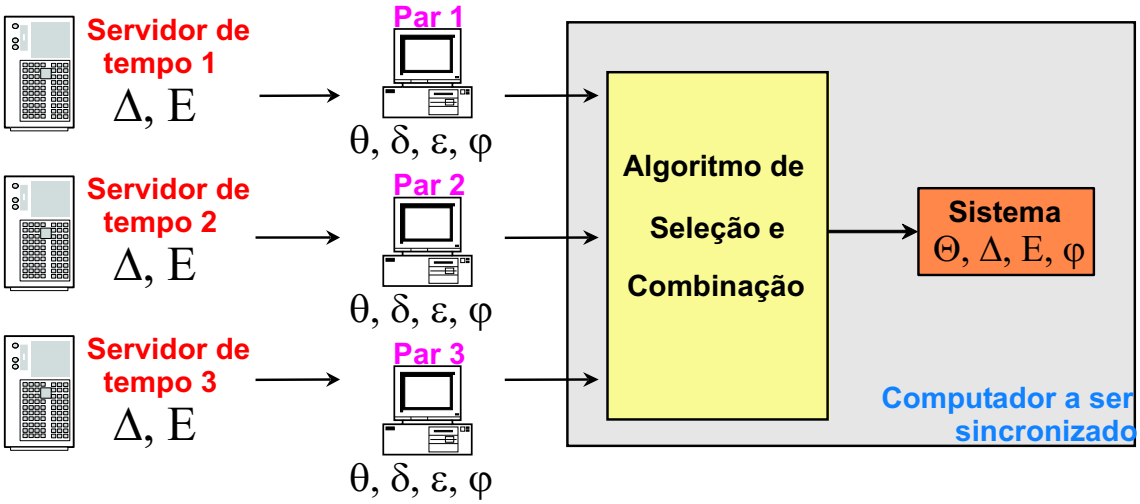

Figura 4.5: Variáveis calculadas para o sincronismo NTP.

Cada par executa o cálculo de seu tempo independentemente dos intervalos de apuração determinados pelo sistema ou pelo servidor remoto. Todo servidor de tempo calcula suas variáveis de compensação Θ, atraso Δ e dispersão E relativas à raiz

da sub-árvore de sincronização, conforme ilustra a Figura 4.5 (MILLS, 2003c).

Para cada mensagem NTP que chega, o par ou servidor de *stratum* menor processa a atualização de suas variáveis de compensação θ, atraso δ, dispersão ε e oscilação φ, do carimbo de tempo e algoritmo de filtro de relógio. No computador que deseja sincronizar seu relógio o sistema de intervalos de apuração, a seleção do relógio e algoritmos de combinação atualizam as variáveis do sistema Θ, Δ, E e φ. As dispersões ε e E aumentam com o tempo dependendo da tolerância de freqüência especificada ϕ (MILLS, 2003c).

A compensação θ_0 é medida através do mais baixo atraso δ_0. A Figura 4.6 ilustra o cálculo do atraso e da compensação. A métrica de distância de sincronismo λ é baseada no atraso, tolerância de freqüência e o tempo desde a última medição.

Para ajustar o relógio de um computador, o NTP pode levar alguns minutos como também algumas horas. Isso ocorre porque o cálculo da média de latência dos vários servidores de um cliente e a escolha do melhor tempo, como foi descrito acima, não é trivial. Geralmente isto acontece em 5 minutos.

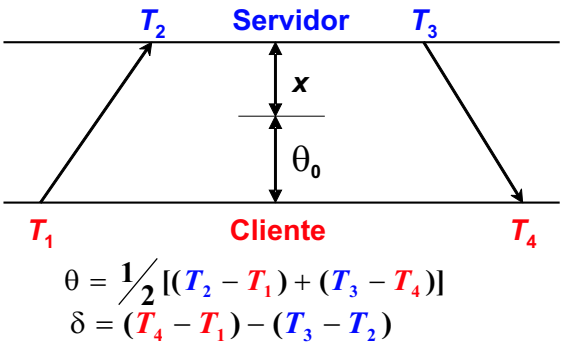

Figura 4.6: Cálculo da compensação e do atraso do tempo no NTP.

Para permitir uma maior precisão e para que o objetivo de sincronizar relógios seja alcançado, o NTP utiliza um sistema que faz com que grandes ajustes ocorram rapidamente e que pequenos ajustes ocorram mais lentamente. Para pequenas diferenças, o NTP utiliza ajustes graduais chamados de “retorno” (*slewing*) e para grandes diferenças de tempo o ajuste é imediato e é chamado de “quebra” (*stepping*) (DEETHS; BRUNETTE, 2001a). Os diferentes tipos de ajustes são mostrados na Figura 4.7.

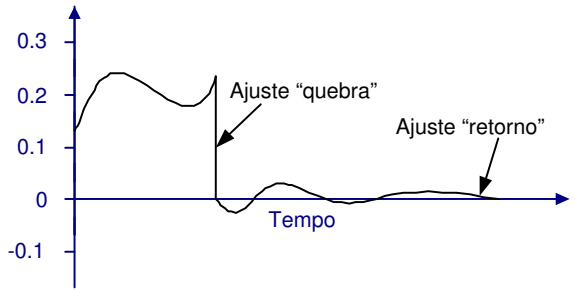

Figura 4.7: Ajustes utilizando os métodos da “quebra” e “retorno”.

Como mostrado na Figura 4.7, quando há uma diferença muito grande entre o tempo do relógio do computador cliente e o tempo do servidor escolhido, há um ajuste brusco de forma a sincronizar o relógio do cliente com o horário correto. Caso esta diferença não seja muito grande, o ajuste é feito gradativamente, até que o relógio do cliente fique sincronizado com o do servidor.

Cabe ressaltar que a precisão do relógio de um cliente depende de sua fonte de tempo. Se o cliente possui apenas uma fonte de tempo e esta for imprecisa, conseqüentemente o tempo do cliente também será impreciso.

4.5 Segurança

O NTP trabalha com várias camadas de segurança, onde se encontram informações protegidas com criptografia simétrica, assimétrica e informações não protegidas. Carimbos de tempo e informações temporais são consideradas de domínio público e não são protegidos.

Modelos de autenticação convencionais são baseados em Infra-estrutura de Chaves Públicas (ICP) e assinaturas digitais. Entretanto, para a distribuição de tempo seguro, estes modelos não podem ser utilizados da mesma maneira que são utilizados na criptografia convencional, devido ao processamento das funções, visto que se está trabalhando com tempo e que o tempo para o processamento de funções de criptografia é variável. Estes cálculos também seriam inviáveis para um servidor que atende vários clientes ao mesmo tempo, pois requereria um grande poder de processamento.

Um cliente NTP utiliza diversos servidores de tempo. O cliente descobre servidores de tempo através de outros serviços que um determinado servidor disponibiliza ou interceptando mensagens que o servidor envia de modo *multicast*. Nos modos ponto a ponto e cliente/servidor um cliente envia um pedido para o servidor e espera que este envie outra mensagem contendo o valor do tempo adicionado com o atraso da rede. No modo *multicast* o servidor envia uma mensagem para o endereço de IP do seu grupo *multicast* e os clientes pertencentes a este grupo ficam aguardando a chegada de alguma mensagem.

Para o NTP, se a sua hierarquia for segura, os caminhos entre os componentes desta hierarquia também são seguros, ou seja, se o servidor de nível primário (*stratum* 1) estiver corretamente sincronizado e autenticado, e se cada servidor/cliente estiver sincronizado e autenticado com servidores/clientes de *strata* mais baixa, então a hierarquia NTP é considerada segura (MILLS, 1999).

O modelo de segurança de autenticação do NTP é baseado apenas na autenticação do servidor perante o cliente. Um cliente é autêntico se puder verificar a identidade de um dos seus servidores, mensagens enviadas pelo cliente não são modificadas ao longo do caminho e pelo menos um caminho entre o cliente e um dos seus servidores seja autêntico.

A partir da versão 2 do NTP, há a preocupação com a autenticação dos elementos pertencentes à hierarquia NTP. Nesta versão havia uma maneira rudimentar de autenticação que se baseava em listas de acesso, ou seja, uma lista contendo os endereços IPs dos clientes autorizados era mantida pelo servidor. Também na versão 2 e nas posteriores 3 e 4, foi incluído um mecanismo mais eficiente de autenticação baseado em criptografia simétrica. Neste esquema, o servidor gera uma lista de chaves simétricas que são distribuídas para os clientes que irão usufruir do serviço prestado. Utilizando os recursos da criptografia, o servidor pode ser autenticado perante o cliente. Para a troca de mensagens, o protocolo determina que o resumo criptográfico (*hash*) do pacote deve ser calculado, cifrado com a chave distribuída pelo servidor e concatenado à mensagem. Assim, as mensagens trocadas entre cliente e servidor podem ter sua autenticidade e integridade verificadas.

Na versão 4 do NTP, há dois esquemas de autenticação: através de chaves simétricas e *Autokey*. A autenticação com chaves simétricas, introduzida na versão 3, utiliza DES ou MD5. As chaves precisam ser distribuídas por um meio seguro fora do protocolo NTP (PALKO, 2001). O protocolo *Autokey* resolve o problema da distribuição de chaves através de técnicas de ICP e o uso de algoritmos como RSA de chave pública e privada, MD5 e a distribuição de chaves através do protocolo Diffie-Hellman (MAURER; WOLF, 2000). O MD5 é utilizado para detectar modificações nas mensagens, RSA para verificar a origem e Diffie-Hellman para gerar um valor secreto comum entre cliente/servidor ou um par. Assinaturas RSA com datações são utilizadas para verificar o código de todos as identidades criptográficas.

4.6 Protocolo *Autokey*

O protocolo *Autokey* é um modelo de segurança encarregado pela verificação da autenticação e integridade de informações. Ele é baseado na combinação da ICP e de valores pseudo-randômicos gerados por uma seqüência de funções resumo de um valor criptográfico envolvendo componentes públicos e privados. O protocolo *Autokey* precisa satisfazer as seguintes condições (MILLS, 2003a):

- Precisa ser compatível com o modelo de arquitetura NTP já existente;
- Precisa determinar independentemente os valores criptográficos e de tempo. Um pacote NTP só é processado quando os valores criptográficos são obtidos e verificados e se o cabeçalho NTP passa pelas verificações de sanidade;
- Não poderá prejudicar a precisão do algoritmo de sincronismo NTP, sua demanda pela rede deve ser mínima, e os recursos de memória e processador devem ser poupados;
- Deve ser resistente a ataques criptográficos;
- Deve possuir uma grande gama de algoritmos criptográficos para não ficar limitado a poucas opções de escolha;

- Precisa funcionar em todos os modos suportados pelo NTP, que são os modos cliente/servidor, ponto a ponto e *broadcast*;
- Deve possuir uma configuração de cliente e servidor clara, principalmente as configurações relacionadas às chaves criptográficas e os certificados digitais;
- A implementação deve ser capaz de gerar arquivos de chaves específicos para cada cliente e servidor.

O protocolo *Autokey* utiliza certificados digitais para a identificação dos servidores. Certificados digitais são documentos eletrônicos assinados por uma Autoridade Certificadora (AC), contendo o nome do proprietário, a entidade emissora, a chave pública do proprietário e o período de validade do certificado. Na implementação da versão 4 do NTP, são utilizados certificados auto-assinados. Um certificado digital auto-assinado é um certificado assinado com a própria chave privada do proprietário do certificado.

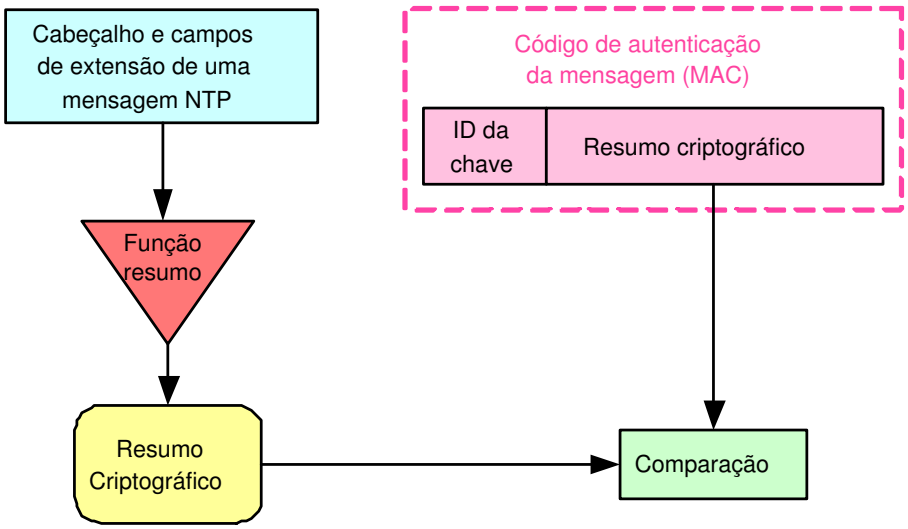

Figura 4.8: Procedimento de autenticação de mensagens.

As versões 3 e 4 do NTP utilizam a função resumo MD5 com uma chave privada de 128 bits e um identificador de chave (*key ID*) de 32 bits (MILLS, 2003a). A autenticação das mensagens, cifradas com criptografia simétrica ou assimétrica, é feita como mostrado na Figura 4.8.

Após o servidor receber uma mensagem de um determinado cliente, ele procura a chave de sessão associada ao identificador da chave (ID de chave) - esta veio com a mensagem recebida pelo servidor - em uma lista contendo todas as chaves de sessão e seus respectivos identificadores. Com a chave, o servidor calcula o resumo do cabeçalho e dos campos de extensão da mensagem recebida. O ID de chave e o resumo da mensagem provenientes do código de autenticação de mensagens (*Message Authentication Code*-MAC) estão anexados na mensagem recebida. O cliente realiza a mesma tarefa que o servidor utilizando a cópia local da chave. Assim, o cliente compara o resultado de seus cálculos com o resumo da mensagem pertencente ao MAC. Se os valores são os mesmos, a mensagem é assumida como autêntica.

Há 3 variações do protocolo *Autokey*, uma para cada modo de operação do NTP, ponto a ponto, cliente/servidor e *broadcast*. Todas as três variações fazem uso de chaves de sessão especiais, chamadas *autokeys* e uma seqüência pseudo-randômica pré-computada de *autokeys* com os identificadores das chaves salvos em uma lista de chaves. O protocolo *Autokey* opera de forma independente para cada associação, ou seja, pode haver várias seqüências *autokey* trabalhando ao mesmo tempo.

Uma chave de sessão *autokey* é formada por 4 campos, como é mostrado na Figura 4.9. Os 4 valores são resumidos pelo algoritmo MD5 produzindo um tamanho de chave de 128 bits (MILLS, 2003a).

Figura 4.9: Estrutura da chave de sessão *autokey*.

Para o IPv4, os campos “IP da fonte” e “IP do destino” contém 32 bits, para o IPv6, estes mesmos campos possuem 128 bits. O ID de chave e o campo de *cookie* contem 32 bits. O endereço IP da fonte e do destino e o ID de chave são valores públicos e estão sem proteção no pacote, enquanto o *cookie* pode ser um valor público ou um valor privado compartilhado, dependendo do modo da associação entre os computadores (MILLS, 2003a).

Para pacotes sem campos de extensão, o *cookie* é um valor privado compartilhado e cifrado. Para pacotes com campos de extensão, o *cookie* tem um valor padrão igual a zero, desde que os pacotes possam ser validados independentemente do uso de assinaturas digitais. Os campos de endereçamento visíveis no pacote transmitido devem ser os mesmos utilizados para a construção da seqüência *autokey* e a lista de chaves. Os campos comuns no pacote transmitido e no pacote recebido devem possuir os mesmos valores (MILLS, 2003a).

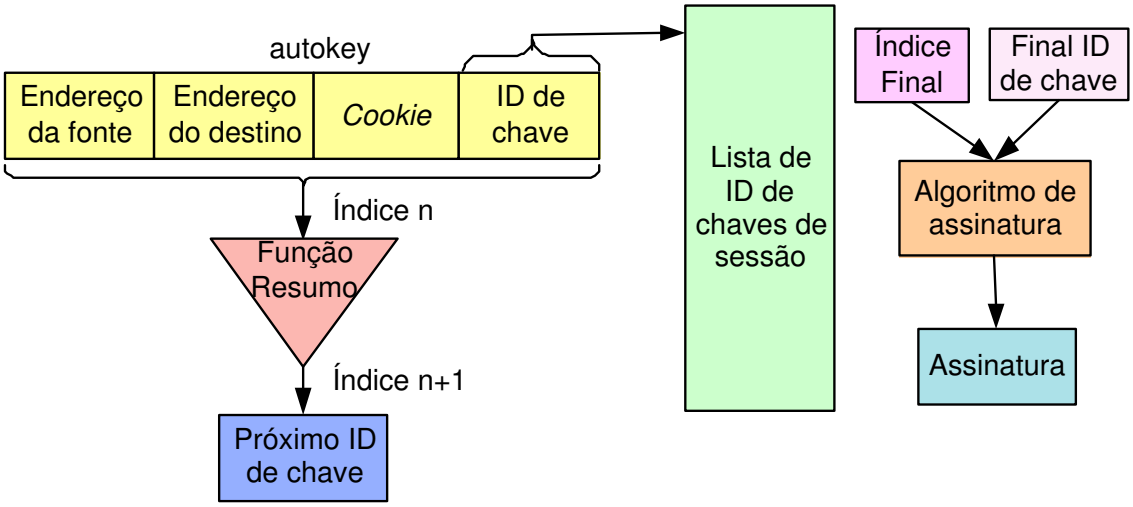

Figura 4.10: Construção da lista de ID de chaves.

A lista de chaves consiste em uma seqüência de IDs de chave, começando com um número randômico de 32 bits (semente *autokey*) igual ou maior que o primeiro ID de chave. A primeira chave de sessão *autokey* é computada como mostrado na Figura 4.10 para a obtenção do próximo ID de chave. A lista de chaves também vai sendo construída. O tempo de vida de cada chave é igual a um intervalo de apuração, ou seja, o intervalo de tempo em que cada cliente precisará se autenticar novamente perante o servidor e quando todas as variáveis relacionadas ao sincronismo são zeradas (MILLS, 2003a).

O índice do último ID de chave na lista é salvo com o próximo ID de chave que chega, coletivamente chamados de valores *autokey*. Estes valores *autokey* são então assinados. A lista é utilizada em ordem inversa como mostrado na Figura 4.11, ou

seja, a primeira chave *autokey* utilizada é a última gerada. Estas chaves são utilizadas na troca de mensagens (MILLS, 2003a).

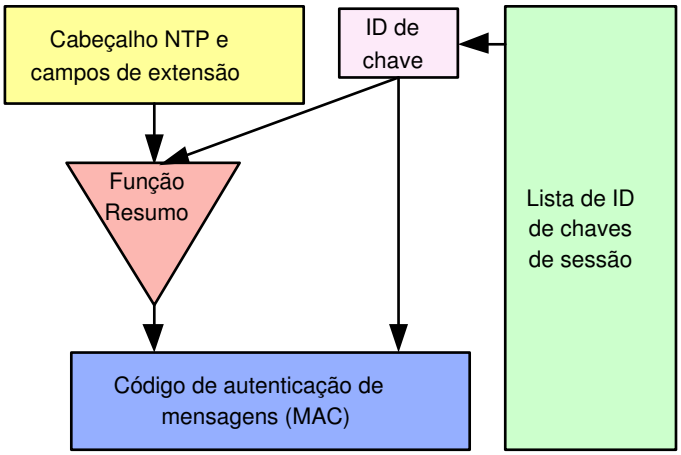

Figura 4.11: Utilização da lista de ID de chaves.

Assinaturas RSA e chaves de sessão são utilizadas em todos os modos. *Cookies* são utilizados em todos os modos e podem assumir muitos valores, dependendo do modo.

O servidor não mantém o estado de cada um de seus clientes, ou melhor, as variáveis pertinentes a cada cliente não são mantidas pelo servidor. Todavia, este utiliza um algoritmo rápido e um valor privado de 32 bits (semente do servidor) para gerar novamente o *cookie* na chegada do primeiro pacote de um de seus clientes. O *cookie* é calculado como sendo os primeiros 32 bits da chave *autokey*, sendo constituída pelo endereço IP do servidor e do cliente, um ID de chave igual a zero e o *cookie* sendo a semente do servidor. O *cookie* é utilizado para o cálculo da atual chave *autokey* por ambos cliente e servidor e é específico para cada cliente separadamente. O servidor cifra o *cookie* com a chave pública disponibilizada pelo cliente (MILLS, 2003a).

O modo cliente/servidor utiliza o *cookie* e cada ID de chave da lista de chaves para devolver a chave *autokey* e gerar o MAC em um pacote NTP. O servidor utiliza os mesmos valores para gerar o resumo da mensagem e verificar se o MAC e a mensagem estão íntegros. Depois da verificação, o servidor gera o MAC para a resposta utilizando os mesmos valores, sendo que a diferença do MAC anterior é a ordem do

endereço da fonte e o endereço do destino. Quando o cliente recebe o pacote do servidor este gera o resumo da mensagem e verifica o MAC no pacote. O cliente também verifica se o ID de chave é compatível com o último enviado, isso evita ataques de repetição com ID de chave antigo (MILLS, 2003a).

No modo *broadcast*, os clientes normalmente não enviam pacotes para o servidor, exceto quando verificam identidades e fazem a calibração da propagação do atraso. Ao mesmo tempo, o cliente obtém os valores *autokey*.

O modo simétrico (ponto a ponto) é semelhante ao modo cliente/servidor e mantém poucos estados entre a chegada de um pacote e a partida da repetição do mesmo. A lista de chaves é gerada e utilizada separadamente para cada par, mas todas são geradas com o mesmo *cookie*. O *cookie* é gerado da mesma forma como no modo cliente/servidor, exceto que neste caso o *cookie* é um número randômico (MILLS, 2003a).

O protocolo *Autokey* inclui também algumas trocas, cada uma com um objetivo específico. Primeiramente, o cliente obtém o nome do servidor (*host name*), os algoritmos que são utilizados para resumir as mensagens e para a assinatura e o esquema de identificação. Recursivamente ele obtém e verifica os certificados digitais após a troca de certificados e verifica a identidade do servidor nas trocas de identidade. Nas trocas de valores o cliente obtém o *cookie* e os valores de chaves *autokey*, dependendo do modo utilizado. Finalmente, o cliente apresenta seu certificado auto-assinado para o servidor para a assinatura na troca de assinaturas (MILLS, 2003a).

A segurança do protocolo *Autokey* é baseada em certificados digitais assinados e na infra-estrutura de certificação. O protocolo *Autokey* constrói um caminho de certificação do servidor primário, que aparentemente tem certificados auto-assinados confiáveis, recursivamente por *stratum*. Cada servidor acumula certificados não duplicados para todas as associações e para todos os caminhos (MILLS, 2003a).

Uma vez sincronizado com uma fonte confiável, o cliente continua com a troca de assinaturas e o servidor age como uma AC assinando os certificados dos clientes. A AC interpreta os certificados como uma requisição de certificado X.509v3, mas verifica se é auto-assinado. A AC extrai o proprietário do certificado ou sujeito (*subject*), o emissor (*issuer*), os campos de extensão e a chave pública e então constrói um

novo certificado com estes dados, com o seu próprio número serial e um determinado intervalo de tempo e assina utilizando sua chave privada. O cliente utiliza o certificado auto-assinado de acordo com suas próprias regras, podendo até ser ACs de outros clientes (MILLS, 2003a).

Uma troca final ocorre quando o servidor tem a tabela de segundos intercalados (*leap seconds*). O cliente faz uma requisição por esta tabela e compara a sua tabela com a enviada pelo servidor. Se a tabela do servidor é mais nova que a sua, então o cliente substitui sua tabela pela tabela obtida do servidor (MILLS, 2003a).

Depois que os valores *autokey* são obtidos, normalmente o modo de associação fica inativo. Se o relógio do cliente recebe um ajuste de "quebra" ou especialmente se for um ajuste de "retorno" (ver Figura 4.7 na página 53), todos os valores de criptografia e tempo são limpos e o protocolo *Autokey* é reiniciado, o que garante que o relógio do cliente e todos os seus dados referentes ao sincronismo são liquidados. Em um intervalo de apuração de aproximadamente 1 dia, as referências de implementação são esvaziadas para todas as associações, todas as assinaturas são atualizadas, o lixo é apagado e a semente do servidor é atualizada.

Para a correção de erros o protocolo *Autokey* possui um registro de alcançabilidade (*reachability register*) e um contador (*watchdog counter*). Em todo o intervalo de apuração o valor do registro de alcançabilidade é deslocado para a esquerda, o bit de mais baixa ordem é apagado e o bit de ordem mais alta é perdido. Ao mesmo tempo, o valor do contador é acrescentado de um. Se uma mensagem que chegou passa por todas as verificações de sanidade e pela autenticação, o bit de ordem mais alta do registro de alcançabilidade é destacado e o valor do contador torna-se zero. Se algum bit do registro de alcançabilidade estiver destacado, o servidor é alcançável, caso contrário ele é inalcançável.

Nas subseções seguintes serão detalhados o protocolo *Autokey* para os modos de sincronismo do NTP cliente/servidor e ponto a ponto respectivamente. O NTP também possui o modo *broadcast* de sincronismo, mas este modo não é interessante para o presente trabalho. Estas subseções são baseadas em (PALKO, 2001).

4.6.1 Modo cliente/servidor

O diálogo começa quando o cliente constrói um pedido de mensagem com um *cookie* utilizando um modelo padrão e envia para o servidor.

O servidor rotaciona 32 bits de um número randômico, o qual o servidor já possui e é utilizado para todos os clientes, e cria um *cookie* com os 4 primeiros octetos do resumo (*hash*) que é mostrado na Equação 4.2. O *cookie* é assinado através do algoritmo RSA e é enviado para o cliente em uma mensagem de resposta.

$$hash = MD5(IP_{cliente}, IP_{servidor}, K_{id} = 0, N_{randomico}) \tag{4.2}$$

O cliente verifica a assinatura utilizando a chave pública do servidor (a maneira como a chave pública foi obtida não está no escopo do NTP). O cliente verifica se o carimbo de tempo é válido e se possui um valor diferente de zero. Se o valor do carimbo de tempo é zero implica que o servidor não está sincronizado e não pode ser utilizado. O cliente salva o *cookie* e gera uma lista de chaves de sessão utilizando o mesmo.

O cliente constrói uma mensagem de requisição sem extensões para enviar ao servidor, calcula o MAC, segundo a Equação 4.3, utilizando uma chave de sessão de sua lista e constrói o campo “Auth”, sendo que Auth.MAC = MAC e Auth.K_{id} = K_{id}.

$$MAC = MD5(K_{sessao}, camposCabealhoNTP) \tag{4.3}$$

O servidor recupera o seu número randômico e recalcula o *cookie*. O servidor verifica o MAC pela reconstrução da chave utilizando K_{id} de $Auth.K_{id}$, o *cookie* gerado e os IPs do cliente e servidor. O servidor utiliza uma chave teste, que é definida pela Equação 4.4, para resumir o cabeçalho do NTP e comparar o resultado com Auth.MAC.

$$K_{teste} = MD5(IP_{cliente}, IP_{servidor}, K_{id}, cookie) \tag{4.4}$$

Assumindo que o resultado da comparação é válido, o servidor constrói uma mensagem de carimbo de tempo para o cliente utilizando um carimbo de tempo do

NTP e um autenticador, Auth.K_{id} = K_{id}, utilizando a K_{id} enviada para o cliente e um Auth.MAC construído a partir da chave K_{id} do cliente, do *cookie* e dos endereços IPs do cliente e servidor, como pode ser visto na Equação 4.5. É importante ressaltar que o servidor não constrói uma lista de chaves, entretanto a chave de sessão é diferente em ambas as direções pois os IPs do cliente e servidor são permutados no cálculo do resumo criptográfico.

$$K_{sessao} = MD5(IP_{cliente}, IP_{servidor}, K_{ID}, cookie) \tag{4.5}$$

O cliente extrai o K_{id} do Auth.K_{id} e verifica se o mesmo é compatível com a K_{id} da requisição. Se for comparado, o cliente calcula a chave de sessão e utiliza a chave de sessão para validar Auth.MAC. Assumindo que todas as verificações estão corretas, o cliente salva o carimbo de tempo, pois esta informação será utilizada posteriormente nos cálculos de ajuste do relógio do cliente.

A Figura 4.12 mostra esta troca de mensagens.

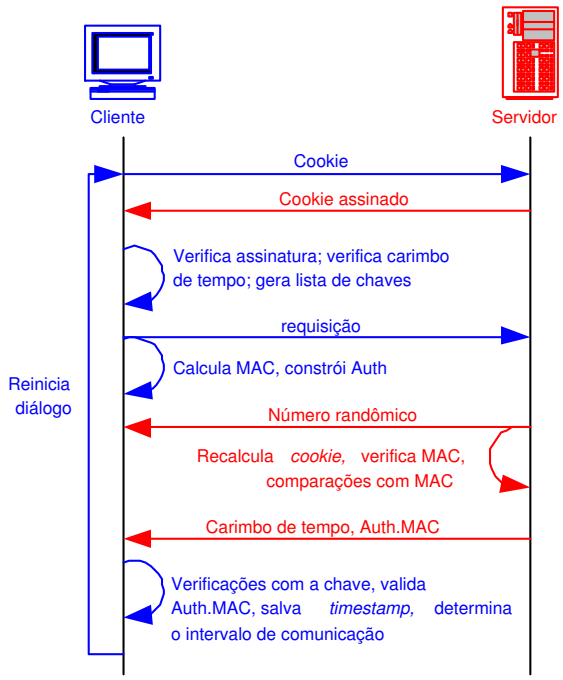

Figura 4.12: Troca de mensagens entre o cliente e servidor no protocolo *Autokey*.

Após o cliente determinar o intervalo de comunicação, este reinicia o diálogo para obter outro carimbo de tempo do servidor.

O diálogo continua neste ciclo até que a lista de chaves do cliente termina. Neste ponto, outra lista de chaves é criada pelo cliente e o diálogo reinicia.

Se alguma autenticação por parte do servidor falhar, este envia um NAK (Reconhecimento Negativo - *Negative Acknowledgement*) como resposta para o cliente, ou seja K_{id} é igual a zero. O cliente ignora a mensagem com K_{id} = NAK e o protocolo será reinicializado.

4.6.2 Ponto a Ponto

No modo ponto a ponto de sincronismo, um computador pode atuar tanto como cliente quanto como servidor. Este modo possui dois submodos: ativo/ativo e ativo/passivo.

No sub-modo ativo/ativo o diálogo é curto, pois o relacionamento entre os pares já foi pré-definido, cada um possui a chave pública de seu par para compartilhamento de mensagens, o nome (*hostname*) de seu par e todos os parâmetros do protocolo de troca de chaves Diffie-Hellman (DH) disponíveis.

Já no sub-modo ativo/passivo o diálogo começa com o par ativo. Estes precisam trocar seus nomes (*hostname*), chaves públicas RSA e configurar os parâmetros para o protocolo DH. O diálogo, no modo ativo/passivo, inicia quando o par ativo obtém todos os parâmetros necessários para o diálogo citado acima. Com estes dados, o par ativo constrói uma lista de chaves utilizando uma chave padrão e um *cookie* e envia sua chave pública para o par passivo com um campo de extensão contendo uma assinatura RSA. O par passivo obtém as informações sobre par ativo através de pedidos de mensagem para este ou de um servidor de certificados confiável. Após receber a chave pública do par ativo, o par passivo verifica a assinatura do pacote enviado comparando a assinatura do mesmo e a chave pública do par ativo que o par passivo já possui e utiliza os parâmetros do protocolo DH para construir uma lista de chaves de sessão, calcula o segredo compartilhado e constrói um *cookie* com os 4 primeiros octetos do segredo. Após estes dados calculados o par passivo envia para o par ativo uma resposta contendo os valores públicos DH, um carimbo de tempo e um pedido das chaves *autokey* nos campos de extensão

assinados.

O restante do diálogo é o mesmo para os sub-modos ativo/passivo e ativo/ativo. Para um melhor esclarecimento, os computadores do par serão chamados de par 1 e par 2.

O par 1 calcula um *cookie* comum utilizando a chave pública DH do par 2 e constrói uma nova lista de chaves utilizando o *cookie* enviado pelo par 2, que são os 4 primeiros octetos do segredo compartilhado. Constrói também um carimbo de tempo e assina as chaves *autokey* e envia para o par 2.

O par 2 constrói uma lista de chaves utilizando o *cookie* comum, envia as chaves *autokey* assinadas e datadas para par 1 e seta um bit de autenticação em sua palavra de controle de estados (*status word*). O par 1 verifica o carimbo de tempo e a assinatura recebidas, armazena as chaves *autokey* e seta um bit de autenticação em sua *status word*.

Neste momento os dois pares podem trocar mensagens para obter dados para o cálculo do atraso da rede. O par que possui um *stratum* menor se torna o servidor e o outro par o cliente. O cliente então sincroniza seu relógio com uma seqüência de requisições de sincronismo.

O ciclo de repetições começa quando o par 1 constrói uma mensagem de requisição para enviar ao par 2 utilizando uma chave de sessão de sua lista para o campo “Auth” (segundo as equações 4.6 e 4.7). O par 2 verifica o MAC primeiramente extraindo a K_{id} de Auth.K_{id} e recria a chave de sessão do par 1, conforme Equação 4.8. Ele confirma que a K_{id} recebida é a próxima esperada do par 1 e utiliza a K_{sessao} para cifrar o cabeçalho NTP recebido, que é calculado segundo a Equação 4.9, e compara o resultado com Auth.MAC.

$$Auth.K_{id} = K_{id} \tag{4.6}$$

$$Auth.MAC = MD5(K_{sessao}, camposCabealhoNTP) \tag{4.7}$$

$$K_{sessao} = MD5(IP_{cliente}, IP_{servidor}, K_{ID}, cookie) \tag{4.8}$$

$$resumo = MD5(K_{sessao}, camposCabealhoNTP) \tag{4.9}$$

Assumindo que o resultado da comparação é válido, o par 2 constrói e envia uma mensagem de resposta para o par 1 incluindo um carimbo de tempo NTP e um MAC construído com a próxima chave de sessão da lista de chaves (segundo equações 4.6 e 4.7).

O par 1 verifica a mensagem recebida da mesma maneira como o par 2 verificou, descrito acima. Assumindo que todas as verificações estão corretas, o par 1 salva o carimbo de tempo. Como no modo cliente/servidor, o carimbo de tempo é utilizado nos algoritmos de obtenção do melhor tempo do NTP, para determinar a capacidade de adaptação dinâmica do par 2 como uma fonte de sincronismo baseado nos cálculos da dispersão e difusão, depois da compensação do atraso de rede.

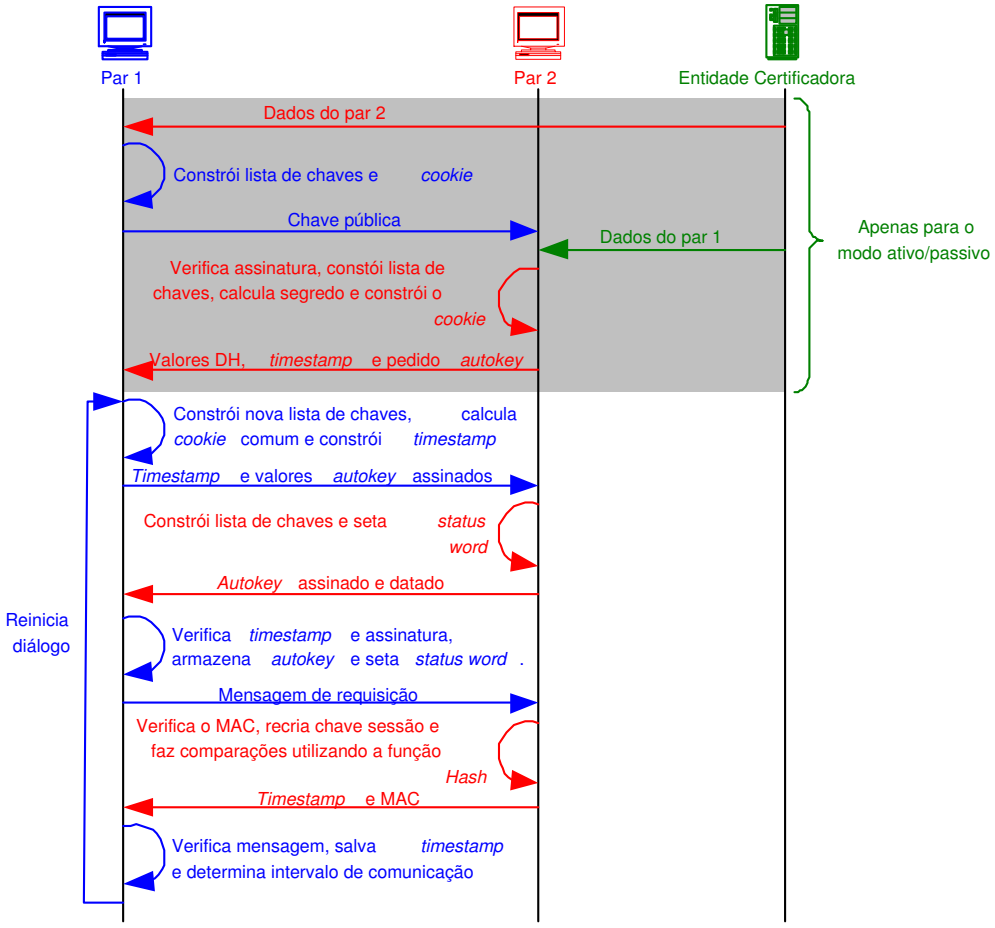

Figura 4.13: Troca de mensagens entre pares no protocolo *Autokey*.

Após o par 1 determinar o intervalo do diálogo, o par 1 reinicia o diálogo

para obter outro carimbo de tempo do par 2. O diálogo continua até que a lista de chaves de um par terminar. A Figura 4.13 exemplifica este diálogo entre pares.

Em caso de perda da mensagem *autokey* ou atualização da chave pública DH por um dos elementos do par, o par prejudicado poderá falhar na autenticação e retornar uma resposta de NAK (K_{id} igual a zero). Assim, a autenticação do par irá falhar.

O protocolo requer que quando um dos elementos do par recebe um valor da chave pública DH que resulte em um *cookie* diferente, ele precisa responder com seu próprio valor da chave pública de DH.

4.7 Conclusão

Para o sincronismo de relógios através da Internet, há a necessidade de técnicas que garantam que o horário a ser transmitido está correto e que chegará ao seu destino íntegro e preciso. Deve-se também garantir que a fonte que está transmitindo o horário é realmente quem diz ser.

O Protocolo de Tempo em Rede - NTP versão 4 supre estas necessidades através de algoritmos que realizam cálculos estatísticos e algoritmos que minimizam os erros e imprecisão do tempo que porventura podem aparecer devido ao translado do tempo através da rede. O NTP, através de técnicas de ICP, garante que o servidor de tempo é uma fonte confiável. O NTP organiza uma topologia de rede que favorece o sincronismo de relógio entre computadores e provê uma precisão consideravelmente boa, considerando-se as condições da rede e o servidor de tempo.

Devido à preocupação com a precisão do tempo e com a segurança das mensagens trocadas no processo de sincronismo, o NTP foi escolhido para fazer o sincronismo do relógio dos componentes da Infra-estrutura de Protocolação Digital de Documentos Eletrônicos.

Capítulo 5

Melhorias no NTP

5.1 Introdução

Abordada toda a parte teórica necessária para a melhor compreensão desta dissertação, agora é hora de realmente tratar das contribuições do presente trabalho, mostradas nos capítulos 5 e 6.

O principal objetivo deste trabalho é a concepção de uma infra-estrutura de protocolação digital de documentos eletrônicos. Justamente por se tratar de tempo, uma infra-estrutura como a proposta deve garantir que os relógios das entidades envolvidas, principalmente os relógios das Autoridades de Datação (ADs), estejam sincronizadas com o horário oficial da nação em que a infra-estrutura esteja operando. Para tanto, como já observado em capítulos anteriores, vários protocolos desempenham o papel de sincronismo de relógios em computadores. Entretanto, o mais adequado para suprir as necessidades da infra-estrutura aqui proposta é o Protocolo de Tempo em Rede (*Network Time Protocol* - NTP), o qual está descrito com detalhes no capítulo 4.

Apesar do NTP possuir um bom desempenho na precisão do tempo distribuído no sincronismo, em relação à segurança este protocolo deixa um pouco a desejar. Como visto no capítulo 4, em relação à autenticação entre clientes e servidores de tempo, o NTP possui autenticação apenas do servidor de tempo perante o cliente, argumentando que é necessário que apenas o cliente saiba da proveniência do tempo que está recebendo,

mas o servidor não precisa saber para qual cliente está enviando seu tempo. Na versão 4 do NTP, esta autenticação já pode ser feita através de certificados digitais auto-assinados.

O fato do certificado do servidor de tempo ser auto-assinado pode causar desconfiança em clientes que desejam receber o tempo de servidores que realmente eles confiem, pois sendo o certificado auto-assinado, torna-se mais fácil algum servidor agir de forma maliciosa e se passar por qualquer outro servidor de tempo. Sendo o certificado digital do servidor emitido por uma Autoridade Certificadora (AC) em que o cliente de tempo confie, fica mais difícil, senão impossível, algum servidor se fazer passar por outro.

Da maneira como o NTP está implementado atualmente, não é possível fazer um rastreamento do tempo, saber se o tempo fornecido a um cliente está sendo utilizado de maneira correta e para onde está indo o tempo que um determinado servidor forneceu. Este tipo de informação pode ser útil para transações que dependem do tempo e também para servidores de tempo que desejam monitorar e rastrear o tempo ou cobrar pelos serviços prestados.

Tendo em vista esta deficiência de segurança do NTP, este capítulo propõe algumas melhorias que, acrescentadas ao NTP, irão adicionar funcionalidades ao protocolo. A proposta descrita neste capítulo foi publicada em dois artigos (DIAS; CUSTÓDIO; DEMÉTRIO, 2003; DIAS, 2003).

Primeiramente, na seção 5.2 é explicado como funcionam a autenticação do cliente e do servidor com certificados digitais emitidos por ACs externas - autenticação mútua - e a auditoria. A seção 5.3 trata da análise formal do protocolo implementado, através de Redes de Petri e lógica GNY.

O auditor faz a auditoria do tempo nas entidades pertencentes à rede NTP, podendo assim perceber se alguma entidade, estando as entidades devidamente autenticadas, está agindo de forma maliciosa. Através da autenticação com certificados digitais do cliente e servidor, o auditor é capaz de rastrear o tempo.

A seção 5.4 refere-se à conclusão do capítulo.

A implementação da solução encontrada para que ambos, cliente e servidor, sejam autenticados um perante o outro utilizando certificados digitais emitidos por

uma AC externa e o programa auditor são mostrados com detalhes no apêndice B.

5.2 Autenticação Mútua e Auditoria

Para que a autenticação mútua e a auditoria fossem possíveis, houve a necessidade de adicionar novos elementos que antes não estavam presentes na rede de sincronismo NTP, os quais estão listados abaixo:

Auditor: Entidade responsável por supervisionar os relógios das entidades envolvidas no sincronismo através do NTP;

Autoridade de Datação (AD): Responsável por protocolar documentos eletrônicos. Há uma AD específica para protocolar os arquivos de *log* do auditor e dos clientes de tempo (podendo ser outras ADs);

AC/LCR: A Autoridade Certificadora (AC) é responsável por emitir os certificados digitais utilizados para a autenticação das partes; a Lista de Certificados Revogados (LCR) é responsável por armazenar e divulgar os certificados digitais já revogados;

Fonte de tempo: Fonte confiável pelo auditor e pela AD supracitada, esta fonte fornece o tempo que será utilizado como base de comparação para outros tempos.

A Figura 5.1 ilustra os elementos que foram adicionados ao NTP.

O auditor não deverá ter sua identidade revelada para que as entidades auditadas não tenham como agir de forma maliciosa perante o auditor, enviando um tempo diferente daquele que esta entidade esteja utilizando. As entidades sujeitas à auditoria saberão que há um auditor entre eles mas não saberão quem é exatamente o auditor, pois este possuirá sua identidade camuflada por seu hospedeiro, podendo variar de tempos em tempos. A requisição de tempo feita pelo auditor será, na realidade, feita por seu hospedeiro, utilizando o certificado digital do mesmo para a autenticação.

O auditor deverá salvar todas as respostas enviadas pelas entidades sob auditoria relativas às solicitações feitas por ele em um arquivo de *log* e após um determinado período de tempo (este período é flexível, podendo ser estipulado pelo administrador

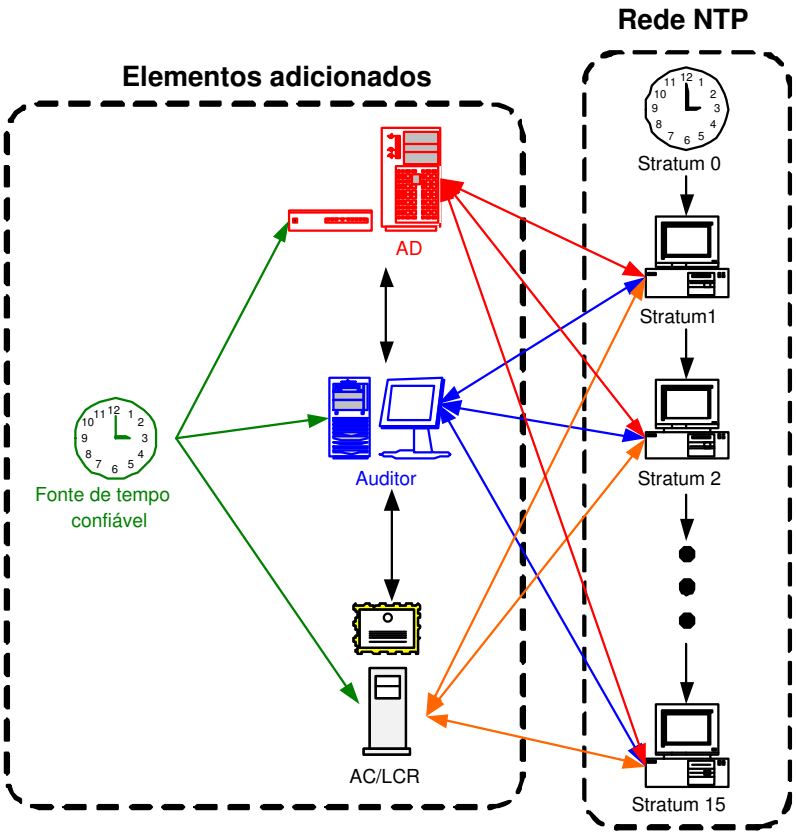

Figura 5.1: Elementos adicionados à rede NTP.

do auditor) o auditor envia o resumo deste *log* para a AD para que seja protocolado. Esta medida de segurança evita dúvidas quanto à integridade do auditor. Esta AD deve utilizar a forma de datação de documentos eletrônicos híbrida, pois esta forma de datação agrega uma maior confiabilidade à datação, visto que não depende apenas do relógio da AD estar sincronizado e também não depende apenas do encadeamento dos resumos dos documentos datados, a datação do documento pode ser verificada de duas formas. A AD deve protocolar os arquivos de *log* provenientes do auditor e eventualmente os *logs* das entidades auditadas, se estas desejarem.

Após a verificação de que o tempo de uma certa entidade está de acordo com padrões pré-configurados, o auditor emite um “alvará” que possui um certo período de validade. Este alvará é assinado pelo auditor ou pela organização que faz a auditoria. O alvará garante que durante o seu período de validade o relógio da entidade auditada está sincronizado com uma fonte de tempo segura e confiável. A validade do alvará emitido

pelo auditor pode ser de algumas horas, um dia ou o período que a organização que faz a auditoria achar necessário.

A fonte de tempo fornece seu tempo tanto para o auditor quanto para a AD. O auditor sincroniza seu relógio com a fonte de tempo para que os tempos das entidades auditadas sejam comparados com o tempo do próprio auditor. A fonte de tempo não precisa, necessariamente, estar diretamente ligada ao auditor; basta o auditor estar sincronizado a uma fonte de tempo segura em que tenha confiança.

A AC fornece os certificados digitais para os membros que desejam fazer sincronismo utilizando certificados emitidos por uma AC externa. A LCR é muito importante pois é nela que as entidades verificam se o certificado de uma outra entidade já foi revogado ou não.

Com a auditoria e a autenticação mútua através de certificados digitais emitidos por AC externa, pode-se saber se há algum problema ou se alguma entidade está agindo de forma maliciosa na rede de sincronismo.

No caso de suspeita de mau funcionamento, o auditor é capaz de enviar uma mensagem à entidade auditada avisando que seu tempo está incorreto ou o próprio auditor pode acertar o tempo da entidade em questão, pois ele tem o poder de acertar o relógio das entidades auditadas através de comandos do protocolo NTP. Se após um determinado número de checagens o auditor perceber que esta entidade ainda está trabalhando com um tempo errôneo, o auditor solicita a revogação do certificado digital desta entidade à AC (através da Autoridade de Registro - AR). No caso de uma Autoridade de Datação (AD), será revogado o certificado digital que habilita a AD protocolar documentos eletrônicos.

Assim como o auditor, qualquer entidade envolvida pode solicitar que seu arquivo de *log* - contendo todas as respostas enviadas ao auditor - seja protocolado. Assim, em caso de disputa, o auditor não poderá acusar uma entidade de ter agido de forma maliciosa ou mesmo modificar alguma de suas respostas, visto que a entidade que foi auditada também protocolou seu *log*.

5.3 Validação Formal

Segundo Meadows (1994), encontrar as falhas de um protocolo criptográfico não é uma tarefa trivial e utilizar somente análise informal também não é suficiente. Protocolos extensivamente analisados apenas informalmente apresentaram-se falhos. Por exemplo: Burrows, Abadi e Needham (1990) mostraram que o protocolo ITUT X.509 possui uma vulnerabilidade.

Uma alternativa para provar que um protocolo criptográfico trabalha de maneira correta é a análise formal. A análise formal tem-se mostrado eficiente para apontar falhas que geralmente não são encontradas somente com a análise informal. Segundo Meadows (1994), dentre os métodos formais mais utilizados recentemente destacam-se: os métodos baseados em máquinas de estados, os métodos baseados em lógica e os métodos algébricos.

O auditor foi validado formalmente, primeiramente, com um método baseado em máquinas de estados: Redes de Petri (MURATA, 1989). Garantiu-se assim a confiabilidade do sistema como um todo e seu funcionamento. A Rede de Petri modelada é mostrada na Figura 5.2.

Para a modelagem da Rede de Petri foi utilizado o software *JARP Petri Net Analyzer* [1]. O software JARP gera um arquivo de extensão *.pn* que representa a própria rede. No JARP, a Rede de Petri é desenhada e pode-se fazer a simulação do movimento das fichas na rede.

Para construir o modelo apresentado na Figura 5.2, os seguintes princípios foram observados:

1. O processo de auditoria inicia quando três condições são satisfeitas: (1) há uma entidade que possa ser auditada; (2) o auditor enviou uma requisição de tempo à entidade sob auditoria; (3) o tempo atual está disponível em uma fonte de tempo confiável. Quando estas três condições são satisfeitas, a transição “AuditorColheDados” é disparada e o processo de auditoria inicia;

[1] http://sourceforge.net/projects/jarp/

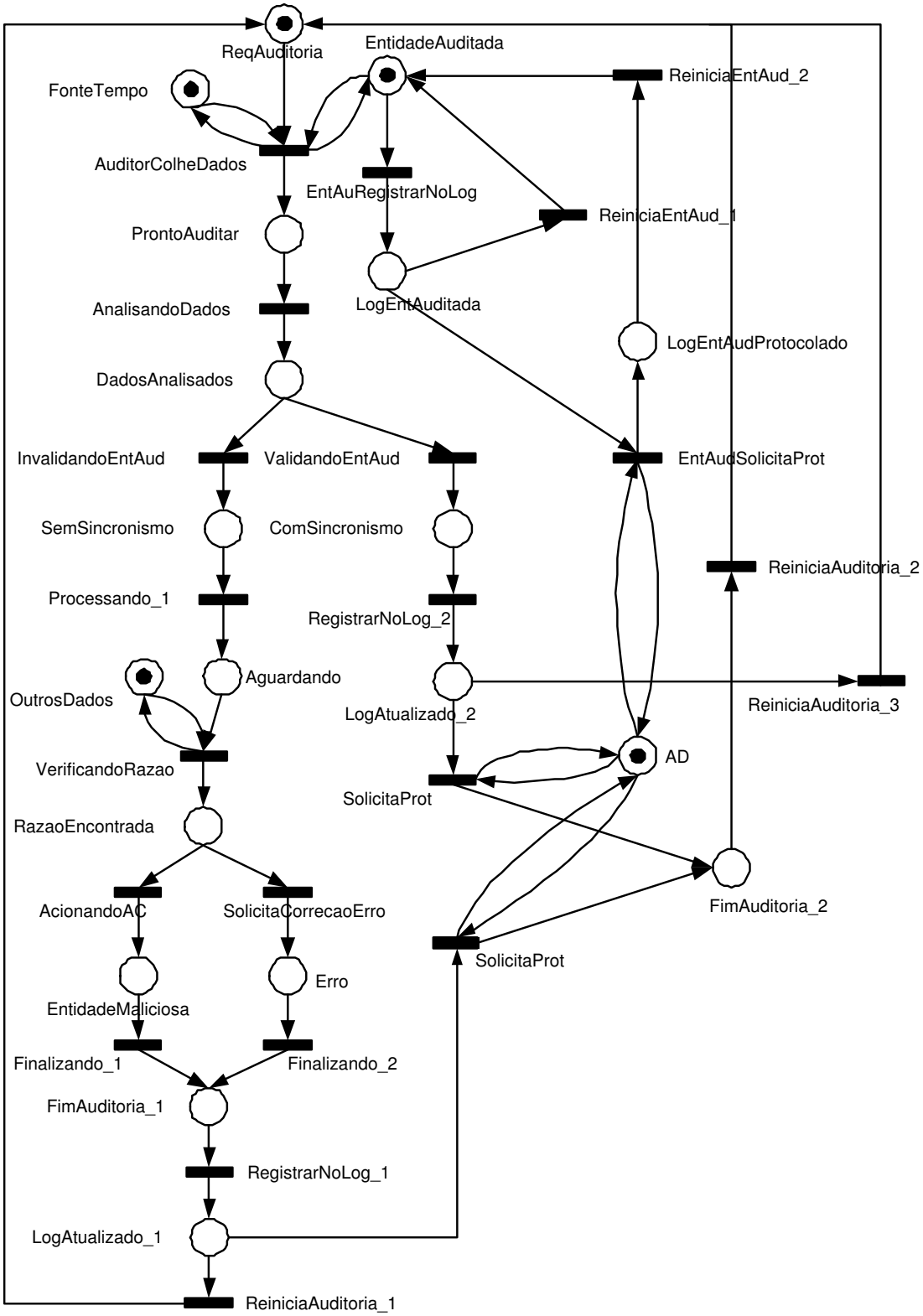

Figura 5.2: Modelagem do auditor com Redes de Petri.

2. Depois do auditor comparar o tempo coletado com o tempo proveniente da fonte de tempo, este dirá se a entidade está ou não sincronizada. Isto é modelado no estado “DadosAnalisados”, onde apenas uma das transições “InvalidandoEntAud” ou “ValidandoEntAud” será disparada;

3. Se a entidade auditada estiver sincronizada com a fonte de tempo confiável, então esta poderá gerar, se a entidade auditada achar necessário, o resumo do seu *log* e enviá-lo para a AD para que seja protocolado. Isto é modelado no não-determinismo do estado “LogEntAuditada”, onde apenas a transição “EntAudSolicitaProt” ou a transição “ReiniciaEntAud_1” será disparada;

4. Se a entidade não estiver sincronizada, duas situações podem ter ocorrido: ou a entidade auditada é maliciosa ou algum erro ocorreu, o estado “OutrosDados” fornece os dados necessários para que o auditor possa saber o que aconteceu. Dependendo da situação, o auditor agirá de maneira diferente. Isto é modelado com o não determinismo que há no estado “RazaoEncontrada”, onde apenas uma das transições “AcionandoAC” ou “SolicitaCorrecaoErro” será disparada;

5. Se for decidido que a entidade é maliciosa, o auditor solicita à AC que revogue o certificado digital desta entidade, representado na transição “AcionandoAC”;

6. Se for decidido que ocorreu algum erro, o auditor toma providências no sentido de corrigir o que ocorreu de errado. Isto está representado no modelo pela transição “SolicitaCorrecaoErro”;

7. Depois do auditor ter finalizado a auditoria, estando a entidade auditada sincronizada ou não com o servidor de tempo do auditor, o auditor poderá protocolar o seu *log* ou poderá continuar a auditoria, isto está representado no não-determinismo dos estados “LogAtualizado_1” e “LogAtualizado_2”. A protocolação do *log* ocorre ao final de uma sessão que pode ser de um dia ou o período que o administrador do auditor desejar.

Para a construção deste modelo, não preocupou-se em validar o sincronismo em si, visto que este é o papel do protocolo NTP. Preocupou-se apenas em validar

e explicar formalmente a função do auditor, ou seja, como é feita a auditoria em relógios de computadores sincronizados pelo NTP.

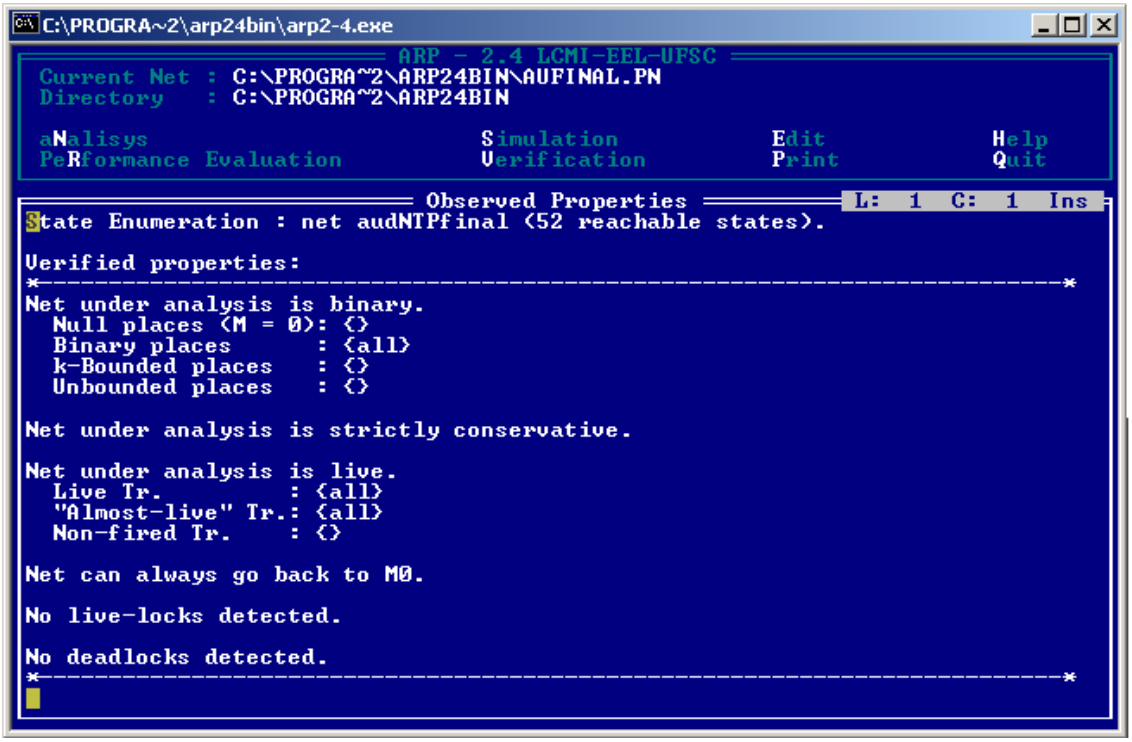

Figura 5.3: Interface da ferramenta ARP com a validação do protocolo.

Para a validação da rede simulada no JARP, foi utilizado a ferramenta ARP [2]. O arquivo *.pn* gerado pelo JARP é utilizado como entrada para a ferramenta ARP, para esta ferramenta fazer a análise da rede. Baseando-se no resultado da análise, como é mostrado na Figura 5.3, chega-se às seguintes conclusões:

- Todos os lugares da rede são binários (há apenas uma ou nenhuma ficha em cada estado), portanto a rede é binária;
- A rede é viva, portanto todas as transições são sempre vivas e não há transição que não dispare;
- A rede é reiniciável, isto é, sempre volta ao estado inicial;

[2]http://www.ppgia.pucpr.br/ maziero/petri/arp.html

- A rede é estritamente conservativa, ou seja, o número de fichas em cada estado permanece constante;
- Não foram encontrados *live-locks* e nem *deadlocks*.

Depois da validação formal do protocolo acima citado com Redes de Petri, o mesmo protocolo foi parcialmente formalizado com um método baseado em lógica: Lógica GNY (GONG; NEEDHAM; YAHALOM, 1990; MATHURIA; SAFAVI-NAINI; NICKOLAS, 1994). Para esta validação formal foi utilizado o *software* SPEAR II (*Security Protocol Engineering and Analysis Resource II*).

O objetivo do SPEAR II é facilitar a engenharia de protocolos criptográficos e ressaltar possíveis erros. Ele possui 4 elementos todos integrados em uma única interface: um ambiente de especificação de protocolo (GYPSIE), uma interface de construção de enunciados GNY, a análise GNY baseada em ProLog (GYNGER) e uma calculadora de rodadas de mensagens. Maiores informações sobre este *software* podem ser encontradas na Internet [3].

Antes de modelar o protocolo no SPEAR, idealizou-se a troca de mensagens envolvidas no protocolo. Para esta tarefa, foi considerada apenas a parte competente ao auditor, ou seja, apenas as mensagens originadas pelo auditor e devolvidas ao mesmo. Também se considerou um fluxo normal do protocolo, sendo assim o auditor não constata qualquer tipo de irregularidade com a entidade auditada. Abaixo está a idealização do protocolo.

Msg 1. A $\Rightarrow$ C: R | N_a | { H(R) | N_a }K_s
Msg 2. C $\Rightarrow$ A: TPc | N_a | { H(TPc) | N_a }K_s
Msg 3. A $\Rightarrow$ P: { H(LR) }KU_p
Msg 4. P $\Rightarrow$ A: TP | { H(LR) }KR_p

Onde:

[3] http://www.cs.uct.ac.za/Research/DNA/SPEAR2/

A: Auditor;

C: Entidade Auditada;

P: PDDE ou AD;

R: Requisição de tempo;

K_s: Chave de sessão compartilhada entre A e C;

H: Função resumo (*Hash*);

Tc: Tempo da entidade auditada;

LR: *Log*;

KU_p, KR_p: Chave publica/privada de P, respectivamente;

Tp: Tempo da PDDE ou AD.

Depois que esta etapa foi cumprida, migrou-se a idealização para a sintaxe do SPEAR II. A Figura 5.4 mostra como ficou a idealização das mensagens do protocolo na ferramenta SPEAR II.

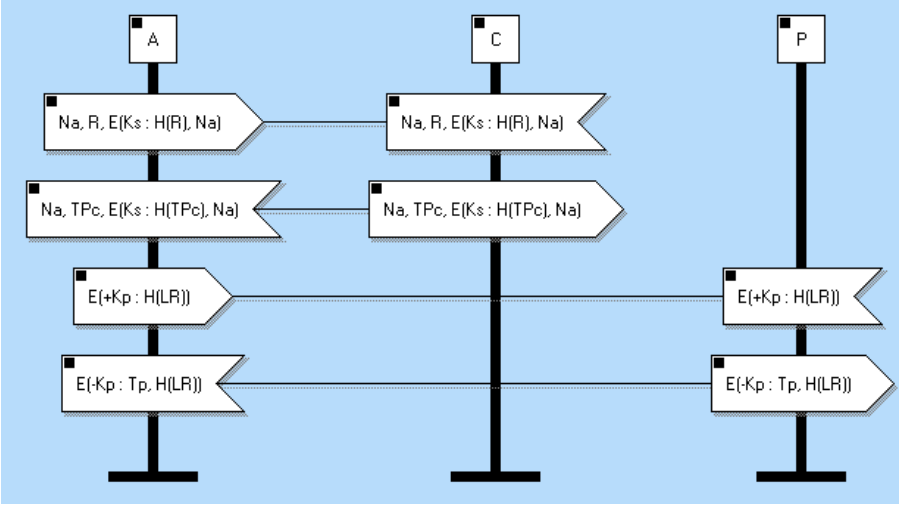

Figura 5.4: Idealização do protocolo no SPEAR II.

Na Figura 5.4, pode-se notar que nas mensagens trocadas entre o auditor e a entidade auditada há sempre um desafio, ou seja, a mensagem que se deseja trocar é enviada sem proteção alguma, concatenada com o resumo desta mesma mensagem cifrada

com a chave de sessão compartilhada entre o auditor e a entidade em questão, para que aquele que recebe a mensagem possa verificar a integridade da mesma. Estas mensagens são mensagens do protocolo NTP.

Para que o SPEAR possa fazer a análise do protocolo é necessário também fazer algumas suposições iniciais sobre os participantes do protocolo, o que eles acreditam e\ou possuem. Há também a necessidade de definir objetivos que devem ser provados pelo *software* para todos os participantes em relação às suas crenças e possessões. A Figura 5.5 mostra um exemplo de suposição inicial.

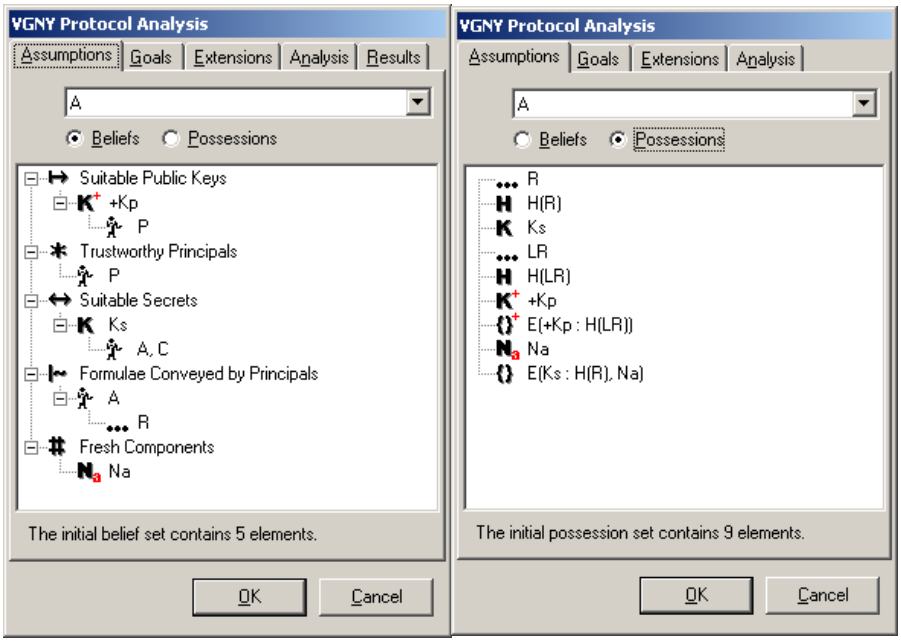

Figura 5.5: Suposição inicial no SPEAR para o auditor.

Na Figura 5.5, o auditor supõe que ele acredita na chave pública da AD, acredita que confia na AD, acredita que há uma chave de sessão compartilhada entre ele e a entidade a ser auditada e por fim, o auditor acredita que foi ele próprio que fez a requisição de tempo. Já nas suposições de possessão, o auditor acredita que possui a requisição de tempo, a chave pública da AD, o resumo da requisição de tempo, o resumo da requisição cifrado com a chave de sessão, a chave de sessão, o seu *log*, o resumo do *log* e o resumo do *log* assinado pela AD. Estas suposições de posse e de crença são feitas para todas as entidades envolvidas no protocolo.

A Figura 5.6 mostra um exemplo dos objetivos a serem provados pelo SPEAR para o auditor. Neste caso, o SPEAR tenta provar através da lógica GNY que há uma chave não comprometida compartilhada entre ele e a entidade auditada e o auditor possui o resumo do seu *log* cifrado com a chave privada da AD ou assinado pela AD.

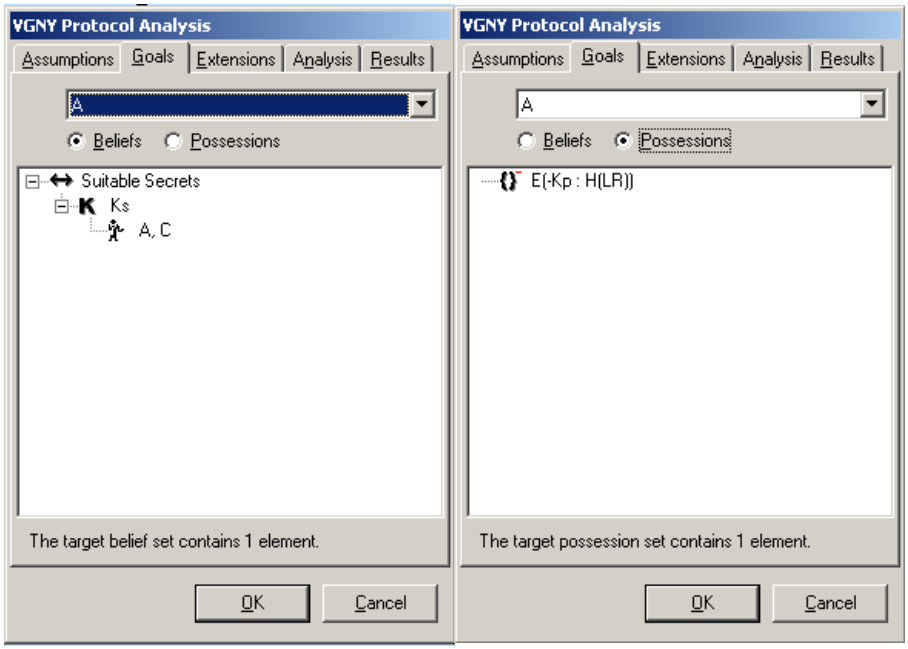

Figura 5.6: Objetivos a serem alcançados no SPEAR para o Auditor.

Após estas configurações, analisou-se o protocolo. O resultado mostrou que todos os objetivos desejados foram alcançados. O SPEAR também gera os resultados em Prolog. Isto possibilita a análise em uma ferramenta Prolog.

Na prática, as mensagens trocadas entre as entidades envolvidas na auditoria utilizam mensagens do protocolo NTP - para as mensagens relacionadas ao tempo - e do Protocolo de Carimbo de Tempo (*Time-Stamp Protocol* - TSP) - para as mensagens relacionadas à protocolação.

5.4 Conclusão

Existem vários protocolos que fazem o sincronismo de relógios de computadores com fontes de tempo. Um dos mais utilizados e o que possui mais funcio-

nalidades é o Protocolo de Tempo em Rede (*Network Time Protocol* - NTP) que faz o sincronismo através da Internet.

Todavia, mesmo com suas diversas funcionalidades, este protocolo deixa a desejar em relação à segurança das partes envolvidas no processo de sincronismo. Assim, este capítulo propõe meios que minimizam falhas de segurança: o auditor e a autenticação mútua.

Foi proposto neste capítulo a autenticação entre clientes e servidores de tempo através de certificados digitais emitidos por ACs externas e também um auditor que monitora o tempo das entidades NTP. Estas melhorias possibilitam a rastreabilidade do tempo na rede NTP e a correção de erros na transmissão ou mesmo no relógio das entidades.

Parte da proposta foi validada formalmente com um método baseado em máquinas de estado - Redes de Petri - e com um método baseado em lógica - Lógica GNY. A validação formal não foi completa, mas através dela pôde-se verificar o potencial da proposta de autenticação e auditoria no NTP.

Capítulo 6

Infra-Estrutura de Protocolação Digital de Documentos Eletrônicos

6.1 Introdução

Como acontece com documentos tradicionais, o documento eletrônico, para ter eficácia jurídica, precisa ser, em suma, assinado e datado. Assim, organizações que utilizam documentos eletrônicos em suas transações e desejam que eles possuam eficácia jurídica, devem utilizar métodos que garantam a assinatura e a datação de forma segura. Infra-Estrutura de Chaves Públicas (ICP) resolve o problema da assinatura digital e a Autoridade de Datação (AD) resolve o problema da datação. Mas uma AD possui uma capacidade limitada em relação ao número de protocolações que pode fazer por minuto, por exemplo, ou em relação ao número de clientes que pode atender. Outro problema é a comparação temporal entre documentos datados em diferentes ADs, visto que nada garante que os relógios das ADs estão corretos (para a forma de datação absoluta) ou se basear apenas no encadeamento dos resumos, nada se pode afirmar para documentos datados em diferentes ADs (para a forma de datação relativa). A solução encontrada para resolver estes problemas foi a concepção de uma infra-estrutura de protocolação digital.

Neste capítulo é proposta uma infra-estrutura de protocolação digital de documentos eletrônicos englobando todos os elementos que a compõem, os protocolos

de comunicação entre os elementos, gerência, auditoria, configuração inicial e o formato do recibo de um documento protocolado em uma AD pertencente a uma infra-estrutura como a proposta neste trabalho.

O primeiro passo da proposta de uma nova infra-estrutura de protocolação digital de documentos eletrônicos é definí-la de maneira a resolver os problemas citados. A definição é apresentada na seção 6.2. Após a definição, é necessário saber como o recibo de um documento protocolado por uma AD pertencente à infra-estrutura proposta atende às novas definições. O formato do recibo é tratado na seção 6.3. A seção 6.4 define os protocolos que fazem com que os elementos pertencentes à infra-estrutura se comuniquem. Como foi criada uma "rede de ADs", originou-se também a necessidade do gerenciamento desta rede, proposta na seção 6.5. Após ter tratado de todos os ítens relativos à infra-estrutura, faz-se necessário explicar o que precisa ser feito quando uma infra-estrutura como esta é implementada em uma organização. A explicação da configuração inicial encontra-se na seção 6.6 . A seção 6.7 destina-se à conclusão do capítulo.

6.2 Definição

A infra-estrutura de protocolação digital de documentos eletrônicos consiste em uma rede heterogênea onde estão presentes elementos necessários para a protocolação de documentos eletrônicos, como Autoridade de Datação (AD), Autoridade Certificadora (AC), Autoridade de Registro (AR), fonte de tempo confiável, gerente, auditor de tempo e clientes de protocolação. Para que a infra-estrutura seja amigável ao usuário (cliente que deseja protocolar documentos), ela deve trabalhar de maneira transparente, ou seja, o usuário apenas solicita a protocolação de um determinado documento e após um curto período de tempo recebe seu documento protocolado, sem se preocupar com qualquer tipo de problema que possa acontecer. Na Figura 6.1 está representada a infra-estrutura de protocolação digital de documentos eletrônicos.

Como pode ser visto na Figura 6.1, os elementos pertencentes à infra-estrutura são os seguintes:

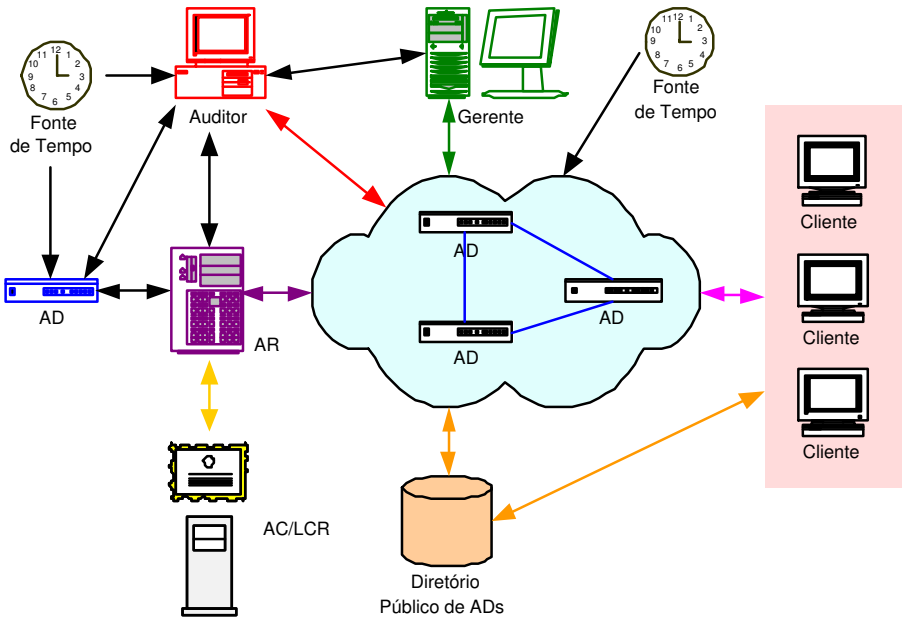

Figura 6.1: Representação da Infra-estrutura.

Fonte de Tempo: Fornece o tempo para todos os outros elementos pertencentes à infra-estrutura. Pelos princípios do protocolo de sincronismo de relógio, um elemento da infra-estrutura pode possuir mais de uma fonte de tempo, desde que seja de confiança;

Auditor: Faz a auditoria do tempo na infra-estrutura através do Protocolo de Tempo em Rede (NTP). É responsável por monitorar o sincronismo do relógio das ADs. O funcionamento do auditor é explicado no capítulo 5;

Gerente: O gerente é responsável por gerenciar a rede de ADs. A gerência é explanada na seção 6.5;

Autoridade de Datação (AD): Uma AD particular protocola documentos provenientes das autoridades que pertencem à infra-estrutura. Aconselha-se que esta AD não faça parte da rede de ADs e deve ser confiável. Esta AD não deve ser auditada pelo auditor que utiliza seus serviços pois ela saberá quem é o auditor;

Autoridade de Registro (AR): A AR faz a comunicação entre a AC e os clientes que desejam serviços de certificação. A AR é responsável por conferir as solicitações e

enviá-las para a AC e responder ao cliente em nome da AC. Maiores informações podem ser encontradas em (IGNACZAK, 2002);

Autoridade Certificadora (AC) e Lista de Certificados Revogados (LCR): A AC é responsável por emitir os certificados digitais das entidades pertencentes à infra-estrutura. A LCR armazena os certificados digitais já revogados;

Diretório Público de ADs: Todas as ADs devem ser cadastradas em um diretório público comum para que cada AD possa escolher com quem irá sincronizar sua cadeia de protocolação. Será melhor explicado ao longo deste capítulo;

ADs que fazem parte da infra-estrutura: Estas ADs formam uma rede de ADs, protocolando documentos de seus clientes e solicitando protocolações para outras ADs através de protocolações cruzadas;

Clientes de protocolação: Clientes que solicitam protocolação de documentos eletrônicos. Para estes, a infra-estrutura é transparente.

Através desta infra-estrutura, é possível comparar documentos protocolados em diferentes ADs baseando-se apenas no encadeamento, para o caso dos métodos relativos de protocolação.

Em se tratando de ADs que utilizam métodos relativos de protocolação e para que documentos protocolados em diferentes ADs possam ser comparados, é necessário que ADs tenham um ou mais pontos em comum em seus encadeamentos, ou melhor, que haja uma **protocolação cruzada**. O ponto em comum entre as ADs é formado em um determinado intervalo de tempo ou rodadas, a ser configurado pelo administrador, e é dado pela aplicação da função F (a mesma utilizada no encadeamento das ADs) ao último *link* da AD em questão com último *link* da AD com quem se deseja fazer a protocolação cruzada. A função F utilizada para a formação da cadeia deve ser a mesma para ambas ADs. A Figura 6.2 ilustra a protocolação cruzada.

Na Figura 6.2, o *link* da rodada R_4, em ambas as ADs, é formado pelo último *link* da própria AD e com o último *link* da AD com quem se está criando o ponto em comum. Nota-se que o *link* R_4 para as duas ADs são exatamente iguais, pois foram

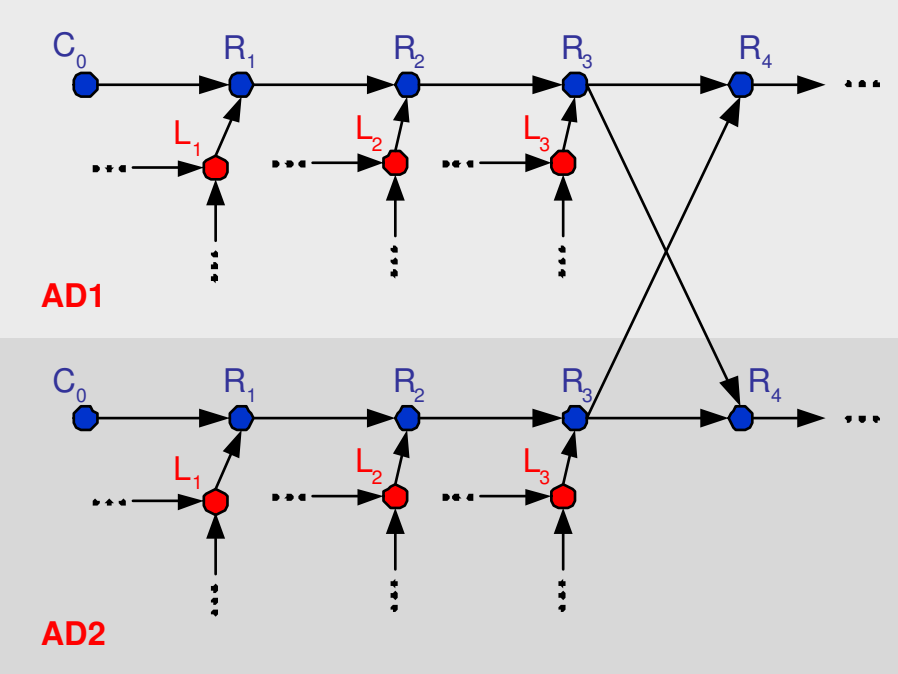

Figura 6.2: Protocolação Cruzada.

formados pelos mesmos pontos, sendo a função *F* responsável por gerar os *links* das rodadas.

Com o ponto em comum, significa que ao percorrer o encadeamento de uma AD, do documento protocolado até o ponto de confiança mais próximo, pode-se percorrer tanto na AD onde o documento foi protocolado quanto em outras ADs com as quais a AD em questão possui pontos em comum no encadeamento.

Dentre duas ADs que fazem protocolação cruzada, em uma delas deve estar configurada a periodicidade da protocolação cruzada, pois quando uma AD receber a requisição de protocolação cruzada, saberá que deve também solicitar uma protocolação para a AD que requisitou anteriormente. Para suportar este novo conceito, foi acrescentada na requisição de protocolação chamada *TimeStampReq* definida na RFC3161 (ADAMS, 2001) em ASN.1 (*Notação de Sintaxe Abstrata Um - Abstract Syntax Notation One*) (ITUT, 1988a, 1988b) um campo *crossedTimeStamp* para identificar que é um pedido de protocolação cruzada e um campo chamado *tsa* para identificar qual a AD que está solicitando a protocolação cruzada, como mostrado abaixo.

Esta nova definição de requisição de protocolação, juntamente com as definições existentes na RFC3161, podem formar outra RFC que abranja também protocolações cruzadas.

```
TimeStampReq ::= SEQUENCE {
        version                 INTEGER v1(1),
        messageImprint          MessageImprint, – Resumo do documento a ser protocolado
        reqPolicy               TSAPolicyID             OPTIONAL,
        nonce                   INTEGER                 OPTIONAL,
        certReq                 BOOLEAN                 DEFAULT FALSE,
        crossedTimeStamp        BOOLEAN                 DEFAULT FALSE,
        tsa                     [0] GeneralName,        OPTIONAL,
        extensions              [0] IMPLICIT Extensions OPTIONAL
}
MessageImprint ::= SEQUENCE {
        hashAlgorithm     AlgorithmIdentifier,
        hashedMessage     OCTET STRING
}
TSAPolicyId ::= OBJECT IDENTIFIER
GeneralName FROM PKIX1 Implicit88 { iso(1)
                identified-organization(3) dod(6) internet(1) security(5)
                mechanisms(5) pkix(7) id-mod(0) id-pkix1-implicit-88(2)
}
```

Quando uma AD recebe um pedido de protocolação com os campos *crossedTimeStamp* e *tsa* preenchidos respectivamente com *TRUE* e o nome de uma AD, a AD que recebeu o pedido deve imediatamente, sem protocolar qualquer outro documento, enviar o *link* da última rodada para ser protocolado na AD que solicitou a protocolação. O campo *tsa* deve ser utilizado apenas em protocolações cruzadas, visto que para protocolações normais este campo viola o requisito de privacidade, pois com este campo preenchido a AD saberá quem está solicitando a protocolação. Estando os dois campos devidamente preenchidos e as ações pertinentes a eles cumpridas, as duas ADs terão um ponto em comum pois possuem um ponto no encadeamento formado pelos últimos *links* de rodada das duas ADs em questão.

Para que as ADs possam escolher com quem fazer a protocolação cruzada, há a necessidade de haver um **Diretório Público de ADs** onde as ADs e dados da mesma, como a função F utilizada, são cadastrados para que o administrador possa escolher com qual AD deseja possuir pontos em comum no encadeamento. No diretório público estão cadastradas todas as ADs em funcionamento, podendo estar organizadas por regiões ou por países, por exemplo. O cadastro deve ser feito a partir do momento que a AD inicia sua operação. Assim, o administrador de qualquer AD deve cadastrar com quais ADs esta irá realizar a protocolação cruzada, caso pertença à uma infra-estrutura. O modelo do diretório público de ADs poderia ser, por exemplo, baseado no modelo do PGP proposto na RFC2420 (CALLAS, 1998).

O diretório público deve ser centralizado, entretanto deve haver cópias deste em vários lugares. Deve ser de fácil acesso para todos, podendo ser um cadastro através da Internet, utilizando o protocolo TCP/IP.

O **gerente** é responsável por gerenciar e manter a infra-estrutura de protocolação, gerando relatórios, calculando dados estatísticos e enviando e recebendo mensagens das ADs sob gerência. O gerente também tem o poder de exigir que uma AD, quando necessário, pare de datar documentos. Isso normalmente acontece quando o auditor verifica alguma irregularidade no relógio da AD.

O **auditor** é responsável pela auditoria do tempo nas ADs, verificando se as mesmas estão utilizando o tempo com a precisão desejável. O auditor se comunica com o gerente e com a AR. Para a AR o auditor irá solicitar a revogação dos certificados digitais das ADs consideradas maliciosas. A identidade do auditor será conhecida tanto pelo gerente quanto pela AR, para que suas funções de auditor sejam bem sucedidas. O funcionamento do auditor está explicado no capítulo 5. As ADs devem ser capazes de fornecer o seu tempo para que o auditor possa auditá-las.

Cabe aqui salientar que as ADs possuem uma chave privada específica para datar documentos eletrônicos, segundo a RFC3161 (ADAMS, 2001). A AD precisa assinar todas as protocolações que esta faz e deve possuir uma distinção entre suas chaves privadas. O certificado relativo a esta chave precisa conter apenas uma instância do campo de extensão do uso da chave, que é definido na seção 4.2.1.13 da RFC2459, com

a variável *KeyPurposeID* tendo o valor *id-kp-timeStamping*. Esta extensão precisa ser crítica. Assim, quando o auditor solicita para a AR que revogue o certificado de uma AD, o auditor se refere ao certificado relativo à chave específica para a protocolação de documentos eletrônicos.

A comparação temporal, considerando-se apenas o encadeamento de dois ou mais documentos pertencentes a uma mesma infra-estrutura, pode não ser uma tarefa fácil. Se os dois documentos foram protocolados por duas ADs diferentes mas ligadas diretamente (possuem protocolação cruzada entre elas) em intervalos diferentes de protocolação cruzada, os documentos podem ser comparados sem problemas, visto que pode-se saber quem foi protocolado primeiro que o outro já que há um ponto em comum no encadeamento entre as ADs. Se os documentos foram protocolados em um mesmo intervalo de protocolação cruzada, apenas pode-se afirmar que eles foram protocolados naquele intervalo. Se as ADs nas quais foram protocolados dois documentos não estejam diretamente ligadas, a busca pelo caminho entre um documento e outro pode ser muito complexa, dependendo de quantas ADs estão entre as ADs que protocolaram os documentos em questão. O gerente pode fornecer uma tabela contendo os relacionamentos entre as ADs, mas para que se busque por documentos, é necessário possuir todo o encadeamento das ADs que protocolaram os documentos e as que estão entre estas duas ADs, ou seja, é necessário ter acesso ao banco de dados interno das ADs. A Figura 6.3 exemplifica a comparação de documentos protocolados em diferentes ADs.

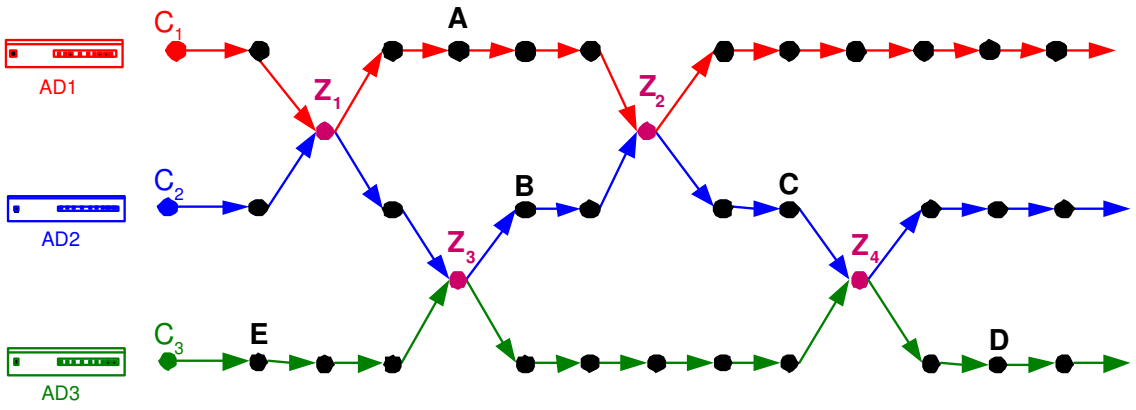

Figura 6.3: Comparação temporal em relação ao encadeamento de documentos.

Na Figura 6.3, para se comparar um documento protocolado em **A** e outro em **B**, nada se pode afirmar observando-se apenas a relação entre eles através do encadeamento, apenas pode-se afirmar que eles foram protocolados naquele intervalo, visto que eles estão entre os pontos de cruzamento $\mathbf{Z}_1$ e $\mathbf{Z}_2$. Já na comparação entre **A** e **C**, pode-se afirmar que **C** foi protocolado depois de **A** pois o ponto $\mathbf{Z}_2$ pode ser considerado como um referencial, já que este ponto está presente nos dois encadeamentos, ou melhor, um documento foi protocolado antes e outro depois de $\mathbf{Z}_2$. A comparação entre documentos da AD_2 e AD_3 acontece da mesma maneira. A comparação entre documentos da AD_1 e da AD_3 é feita através da AD_2. Para se comparar documentos protocolados em **A** e **D**, por exemplo, primeiramente se deve consultar a tabela emitida pelo gerente e ter certeza que a AD_1 está indiretamente ligada à AD_3 e que a AD_2 está ligada com a AD_1 e também com a AD_3. Após esta verificação, deve-se partir de **A** e percorrer a cadeia de protocolação até encontrar a primeira protocolação cruzada da AD_1 com a AD_2 e a partir deste ponto deve-se percorrer a cadeia da AD_2. Procura-se então por um ponto em comum entre esta e a AD_3 e após encontrado este ponto, $\mathbf{Z}_4$, percorre-se o encadeamento da AD_3 até encontrar o documento desejado, neste caso o documento protocolado em **D**. Até mesmo o ponto **E** pode ser comparado com **A**, pois os pontos $\mathbf{Z}_1$ e $\mathbf{Z}_3$ são tomados como referência. Note que para esta comparação, é necessário ter acesso ao banco de dados da AD_1, AD_2 e AD_3.

A seção 6.3 trata do recibo de um documento protocolado em uma AD que faz protocolação cruzada com outras ADs.

6.3 Recibo

O recibo enviado para um cliente que utiliza os serviços da infra-estrutura de protocolação digital de documentos eletrônicos foi criado a partir da RFC3161. O recibo em ASN.1 mostrado nesta RFC trata apenas de datações absolutas, sem qualquer método de encadeamento. Assim, foi acrescentado alguns campos ao recibo proposto na RFC3161 para suprir as necessidades da datação relativa e protocolação cruzada, como é mostrado a seguir. Esta estrutura ASN.1 é uma das contribuições deste trabalho.

```
TimeStampToken ::= SEQUENCE {
        idStamp         INTEGER,
        version         INTEGER v1(1),
        policy          TSAPolicyId,
        serialNumber    INTEGER,
        nonce           INTEGER                         OPTIONAL,
        tsa             [0] GeneralName,
        timeInfo        TimeInfo,
        linkInfo        LinkInfo,
        signature       TSTSign,
        extensions      [1] IMPLICIT Extensions         OPTIONAL
}
TSAPolicyId ::= OBJECT IDENTIFIER
GeneralName FROM PKIX1 Implicit88 { iso(1)
                identified-organization(3) dod(6) internet(1) security(5)
                mechanisms(5) pkix(7) id-mod(0) id-pkix1-implicit-88(2)
}
TimeInfo ::= SEQUENCE {
    genTime         GeneralizedTime,        – YYYYMMDDhhmmss[.s...]Z
    accuracy        Accuracy                OPTIONAL,
    ordering        BOOLEAN                 DEFAULT FALSE
}
Accuracy ::= SEQUENCE {
    seconds         INTEGER                 OPTIONAL,
    millis          [0] INTEGER (1..999)    OPTIONAL,
    micros          [1] INTEGER (1..999)    OPTIONAL
}
LinkInfo ::= SEQUENCE {
    trustedPoint    MessageImprint,
    docHashs        SEQUENCE OF MessageImprint,
    jumpLists       SEQUENCE OF JumpList,
    RoundHash       MessageImprint
}
MessageImprint ::= SEQUENCE {
        hashAlgorithm   AlgorithmIdentifier,
        hashedMessage   OCTET STRING
```

```
}
JumpList ::= SEQUENCE {
        jumpLevel       INTEGER,
        links           SEQUENCE OF Link
}
Link ::= CHOICE {
    roundLink    MessageImprint,          – Rodada
    jumpPoint    MessageImprint,          – Salto
    crossedPoint CrossedPoint             – Protocolação Cruzada
}
CrossedPoint ::= SEQUENCE {
            point           MessageImprint,
            tsa             [0]GeneralName
}
TSTSign ::= SEQUENCE {
        sigAlgorithm    AlgorithmIdentifier,
        signature       OCTET STRING
}
```

Resumidamente, um recibo terá os dados da AD como assinatura, política de protocolação, entre outros dados. Conterá também informações sobre o tempo e sobre a lista de encadeamento. As informações da lista de encadeamento são o ponto de confiança, os resumos dos documentos enviados para protocolação em uma mesma rodada e uma seqüência de *links*. Na seqüência de *links*, deve-se escolher o tipo de *link* que está sendo especificado, se um *link* de rodada, *link* de salto ou *link* de protocolação cruzada. O *link* de protocolação cruzada conterá o resultado da função F dos últimos *links* de rodada das duas ADs que estão fazendo protocolação cruzada.

No recibo de protocolação que retornará para o cliente existe o campo *tsa*, o qual conterá o nome da AD que protocolou o documento. Caso haja protocolação cruzada, o *link* da protocolação cruzada será do tipo *CrossedPoint*, o qual contém o nome da AD com quem foi feita a protocolação cruzada. Assim, o cliente poderá verificar o encadeamento possuindo apenas o recibo da protocolação.

As ADs que utilizam o método da Árvore e o Método da Árvore Sincronizada devem proceder de maneira um pouco diferente em relação à protocolação cruzada, comparando-se com ADs que utilizam outros métodos de datação. A AD1 solicita a protocolação cruzada (enviando o seu último *link*) para a AD2 e fica em modo de espera até receber a resposta da sua solicitação juntamente com a solicitação de protocolação cruzada da AD2.

Como já visto anteriormente, no Método da Árvore Sincronizada a AD faz protocolações em intervalos de tempo ou espera chegar a um determinado número de solicitações de protocolação para atender todas as solicitações e emitir apenas um recibo para os clientes que solicitaram as protocolações em um mesmo intervalo. Entretanto, quando uma destas solicitações for para protocolação cruzada, esta solicitação em especial deve possuir um recibo separado pois tanto a AD1 quanto a AD2 deverão possuir um ponto em comum. Caso fosse processado todas as solicitações juntas como é feito normalmente, os pontos das duas ADs não seriam o mesmo, pois o cálculo de um *link* depende dos resumos de todos os documentos enviados em uma mesma rodada e estes resumos, obviamente, seriam diferentes para ambas.

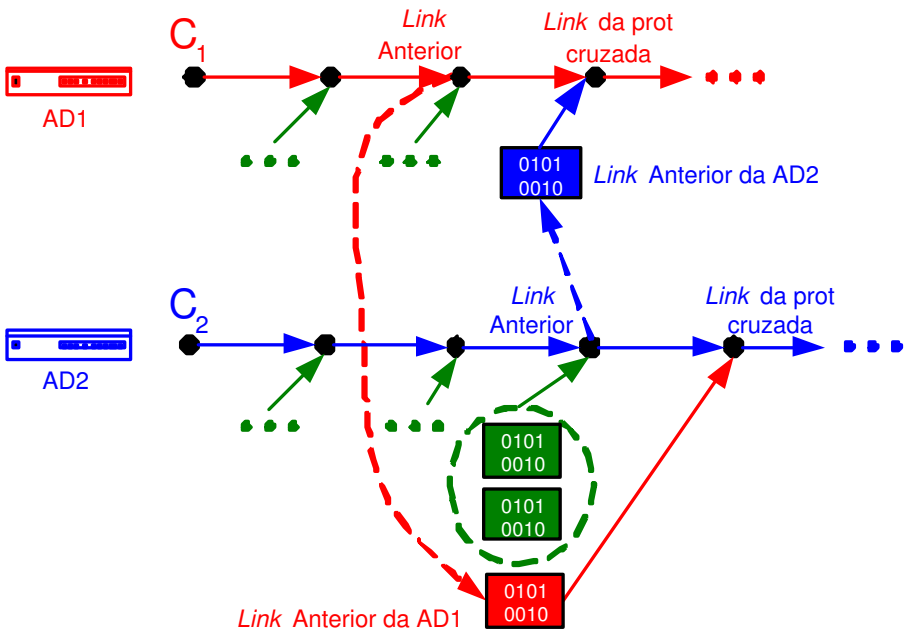

Figura 6.4: Protocolação Cruzada no Método da Árvore Sincronizada.

Após decorrido o intervalo configurado, a AD2 deve verificar todas as solicitações e separar as solicitações de protocolação cruzada. A AD2 deve atender as solicitações comuns, protocolando os documentos e emitindo o recibo aos clientes. Após atender as solicitações comuns, a AD2 irá atender ao pedido de protocolação cruzada, protocolando assim o último *link* da AD1. A AD2 deve também protocolar seu último *link* na AD1, mas como nesse momento o último *link* é o *link* da protocolação cruzada, a AD2 deverá solicitar a protocolação do *link* anterior ao da protocolação cruzada, para que ambas as ADs tenham um *link* que foi formado pelo resultado da função F com entradas iguais. Quando a AD1 recebe a resposta da AD2 e a solicitação de protocolação cruzada, esta imediatamente atende à solicitação da AD2 para depois continuar seu normal funcionamento. Ao solicitar protocolação cruzada para a AD1, a AD2 fica também em modo de espera, como aconteceu com a AD1, a qual iniciou o processo de protocolação cruzada. A Figura 6.4 exemplifica o processo de protocolação cruzada em ADs que utilizam o método da árvore sincronizada.

A análise da confiabilidade e a auditoria do recibo e do encadeamento contendo a protocolação cruzada estão descritos no capítulo 7 do trabalho de Costa (2003).

6.4 Protocolos de comunicação

Protocolos de comunicação são necessários para que os elementos pertencentes à infra-estrutura possam se comunicar e possam ser monitorados. Esta seção tem o objetivo de explicar os protocolos utilizados na infra-estrutura de protocolação digital de documentos eletrônicos.

Todos os elementos da infra-estrutura precisam ter acesso a uma fonte de tempo precisa, íntegra e segura. A comunicação entre a fonte de tempo e todos os outros elementos da infra-estrutura é dada através do Protocolo de Tempo em Rede (*Network Time Protocol* - NTP), já explanado neste trabalho no capítulo 4. Com a adição de funcionalidades ao NTP, vista no capítulo 5, o NTP cumpre todos os requisitos necessários para a obtenção de um tempo preciso aos elementos da infra-estrutura, podendo-se também rastrear o tempo e saber sua proveniência.

O protocolo utilizado para comunicação entre uma AD e outros elementos que necessitam de seus serviços é baseado no Protocolo de Carimbo de Tempo (*Time-Stamp Protocol*- TSP) definido na RFC3161 (ADAMS, 2001). Como já explicado, para satisfazer as necessidades da infra-estrutura, houve a necessidade da modificação do protocolo original descrita na seção 6.2 em ASN.1 relativa à solicitação de protocolação, e também a modificação descrita na seção 6.3 descrita em ASN.1 relativa ao recibo de protocolação.

Para a comunicação entre o auditor e as ADs é utilizado o protocolo NTP acrescido com a funcionalidade de autenticação mútua através de certificados emitidos por Autoridades Certificadoras externas.

O acesso ao Diretório Público de ADs pode ser obtido através do protocolo TCP/IP, pela Internet. O funcionamento do diretório público foi tratado na seção 6.2.

A requisição de certificados digitais é feita da mesma maneira como acontece normalmente, sem modificação alguma. A requisição é entregue a uma Autoridade de Registro que formata um documento com um conjunto de informações. A requisição segue o PKCS♯10 (Padrões para Criptografia de Chave Pública - *Public Key Cryptography Standards*) e contém dados como a chave pública e a identificação do proprietário desta. Pode ser feita através do acesso a uma página WEB ou envio por e-mail da requisição, a qual é processada para gerar o certificado. Maiores informações podem ser encontradas na dissertação de mestrado de Ignaczak (2002).

Para o gerenciamento da infra-estrutura, é utilizado o protocolo PSGPD (Protocolo Simples de Gerenciamento de Protocolação Digital),um protocolo semelhante ao Protocolo Simples de Gerenciamento de Rede (*Simple Network Management Protocol*- SNMP). O PSGPD é descrito com detalhes na seção 6.5.

6.5 Gerência

O protocolo de gerência da infra-estrutura segue o modelo do SNMP, definido em sua primeira versão na RFC1157 (CASE, 1990).

Um programa agente roda nas ADs, trocando informações relativas ao estado da AD com o gerente. O gerente é responsável por gerenciar a rede de ADs, monitorando, controlando as ADs pertencentes a uma determinada infra-estrutura, gerando arquivos de *logs* e emitindo relatórios. O gerente deve trabalhar exclusivamente nas funções de gerenciamento, processando as informações recebidas e gerando estatísticas. O gerente deve estar sempre ativo, coletando informações das ADs. Os agentes nas ADs devem atuar de forma paralela ao processamento das requisições de protocolação para que não haja qualquer comprometimento em relação ao desempenho da AD.

O protocolo utilizado para a comunicação entre agente e gerente é o **PSGPD** (Protocolo Simples de Gerenciamento de Protocolação Digital). O corpo das mensagens trocadas entre agente e gerente no PSGPD pode ser uma das seguintes operações:

- **REQUISIÇÃO-LEITURA**: Utilizado para ler o valor de uma ou mais variáveis, que são informadas na requisição;
- **REQUISIÇÃO-LEITURA-PRÓXIMO**: Utilizado para ler o valor de uma ou mais variáveis que sucedem lexicograficamente às informadas na requisição;
- **REQUISIÇÃO-ALTERAÇÃO**: Atribui valores a uma ou mais variáveis;
- **RESPOSTA-ALTERAÇÃO**: Resposta às operações acima citadas;
- **EVENTO**: Envio de um evento para a estação de gerenciamento. Esta operação é a única que parte do agente.

A Figura 6.5 mostra quais são as mensagens originadas pelos agentes e pelo gerente.

As variáveis que foram mencionadas são os recursos de uma AD que podem ser gerenciados. Uma variável reflete o estado de cada recurso gerenciável, por exemplo se uma AD está ligada ou não. Cada AD possui uma base de dados onde estas variáveis estão armazenadas chamada de **BGIP** (Base Gerencial de Informações de Protocolação). Caso o estado de qualquer recurso gerenciável da AD mude, o agente deve mudar também o estado da variável relativa a este determinando recurso.

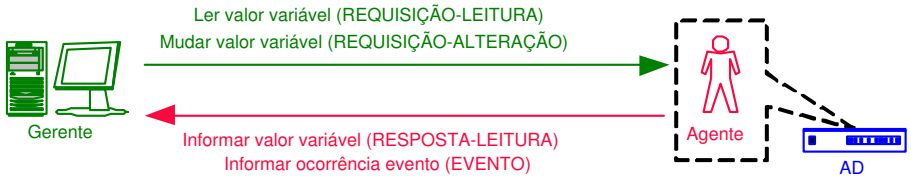

Figura 6.5: Mensagens trocadas entre agentes e gerente.

O corpo da mensagem conterá as operações acima citadas, tendo como parâmetro qual a variável está sendo utilizada e o seu valor. Utilizando as variáveis armazenadas na BGIP, o gerente pode solicitar que a AD execute alguma ação, como parar a protocolação, enviar seu arquivo de *log*, enviar o seu estado atual, entre outras atividades. O agente constantemente está lendo o valor das variáveis da BGIP e caso algum valor é mudado pelo gerente, o agente irá atender ao pedido executando a ação determinada.

Com a operação EVENTO, o agente de uma AD acrescentada à rede informa ao gerente que há um novo elemento a ser gerenciado, por exemplo. A descoberta de alguma AD que se desvinculou da rede é feita pelo gerente, quando este envia requisições para um agente e ele não responde. Após um determinado número de requisições, o gerente considera que esta AD está fora da rede ou que há algum problema com o canal de comunicação, assim o administrador é avisado. O agente é capaz de perceber se algo está errado com a AD e na ocorrência de alguma anormalidade o gerente é informado.

Tanto o gerente quanto os agentes deverão possuir um par de chaves, pública e privada, para cifrarem e decifrarem mensagens trocadas entre eles. As mensagens trocadas entre agentes e gerentes devem trafegar pela rede cifradas através de criptografia assimétrica, utilizando o par de chaves que cada um possui, para a garantia de integridade (dados não devem estar corrompidos ou violados) e autenticidade (apenas o gerente ou um determinado agente tem acesso à informação).

No arquivo de *log* de uma AD deve estar presente o registro de todos os eventos ocorridos na AD. A AD deverá registrar os eventos de forma paralela à protocolação, para que o processamento de requisições de protocolação não seja prejudicado. Os eventos registrados são publicação de ponto de confiança, ocorrência de um salto no encadeamento (se a AD utiliza o Método da Árvore Sincronizada), ocorrência da pro-

tocolação cruzada, adição de ADs na lista de protocolação cruzada, correção do relógio feito pelo auditor, ocorrência de ajuste do relógio, adição ou remoção de fontes de tempo, exportação dos dados da base de dados da AD para outra mídia, expiração do certificado digital, comprometimento da chave da AD e problemas de *hardware*.

O *log* da AD deve ser exportado para uma mídia externa após um determinado tempo ou tamanho do arquivo, sendo configurado pelo administrador da AD. Durante a emissão do *log*, a AD fica indisponível para protocolações, devido a demanda de recursos necessários e devido à utilização do único canal de comunicação entre a AD e o módulo público que fica entre o cliente de protocolação e a AD propriamente dita.

Com a análise do arquivo de *log* das ADs o gerente pode montar uma tabela contendo o cruzamento entre as ADs e gerar um relatório. Assim, quando necessário, o gerente pode disponibilizar esta tabela para uma possível busca de documentos protocolados pelas ADs monitoradas pelo gerente. Isso facilita a verificação realizada nos encadeamentos das ADs envolvidas.

Após o gerente possuir o *log* de uma ou mais ADs, ele faz cálculos estatísticos baseados nos dados coletados. Através da análise dos *logs* e do resultado da análise estatística o gerente tem condições de saber qual AD está com excesso de trabalho comparado com outras ADs de uma mesma infra-estrutura. Sendo assim, o administrador da infra-estrutura é avisado para que providências sejam tomadas caso necessário. O gerente envia *e-mails* para o administrador da infra-estrutura anexando o resultado da análise dos *logs* das ADs ou informa se algo está errado; ele também mantém o resultado da análise estatística para uma posterior consulta.

O gerente também é responsável por analisar se o canal de comunicação entre ele e as ADs não possui problemas, caso contrário, o administrador é avisado.

Um gerente não tem condições de, automaticamente, desviar o tráfico de uma AD para outra. Em outras palavras, o gerente não tem como saber quantas protocolações uma AD possui em um determinando momento para comparar com outra AD. Mesmo que o gerente possua esta informação, não é possível que ele transfira uma requisição de protocolação que foi enviado para uma AD específica para outra.

6.6 Configuração Inicial

Para a infra-estrutura funcionar corretamente, algumas configurações precisam ser realizadas antes de sua operação.

Deve haver fontes de tempo externas disponíveis para que as entidades envolvidas na infra-estrutura possam fazer o sincronismo de seus relógios. As ADs devem sincronizar o seu relógio com o horário oficial da nação onde estas se encontram fisicamente. O administrador ou responsável pela instalação deve ajustar a localização (*time-zone*) de cada AD. Por lei, o Observatório Nacional (ON) é o servidor de tempo oficial do Brasil. Assim, todos os relógios de ADs devem estar sincronizados com o ON. Servidores e clientes de tempo devem ser autenticados um perante ao outro utilizando certificados digitais.

As ADs e os clientes participantes da infra-estrutura devem possuir um certificado digital emitido por uma AC externa para que possa acontecer o rastreamento do tempo, como visto no capítulo 5. As ADs devem possuir um certificado especial que são utilizados apenas para a protocolação de documentos. A requisição de certificado digital é feita para a AC através da AR.

Todas as ADs devem ser cadastradas no Diretório Público de ADs, para que cada AD possa escolher com quem fará o sincronismo do encadeamento, a protocolação cruzada. Os clientes de protocolação também poderão consultar o diretório público de ADs e escolher qual ou quais ADs irão protocolar seus documentos.

Se as ADs utilizam o Método da Árvore Sincronizada (método indicado para ser utilizado na infra-estrutura devido às suas vantagens, visto no capítulo 2), estas devem publicar seu ponto de confiança, caso seu banco de dados ainda não esteja populado. Se o banco de dados estiver populado, basta continuar o encadeamento.

O administrador de cada AD deve configurar com quais ADs uma AD irá fazer a protocolação cruzada e o intervalo entre elas. Dependendo da necessidade, o intervalo entre uma protocolação cruzada e outra não deve ser muito grande. Pois caso haja a necessidade de se fazer a comparação temporal de dois documentos baseando-se apenas no encadeamento, é interessante que o intervalo seja pequeno pois assim a

chance de que dois documentos foram protocolados no mesmo intervalo diminui, já que na comparação de documentos protocolados no mesmo intervalo nada se pode afirmar baseando-se apenas no encadeamento.

Os protocolos de comunicação entre as entidades devem ser devidamente configurados. O *software* de acesso à AD deve estar instalado nos clientes de datação.

Como já detalhado no capítulo 5, o programa auditor deve estar trabalhando em um servidor de tempo já autenticado na rede NTP. Sua identidade está camuflada pelo servidor hospedeiro; exceto para o gerente, que conhece a identidade do auditor.

Após um novo membro ser incorporado na rede de ADs, este deve enviar uma mensagem ao gerente informando seus dados e avisando que agora faz parte da rede. O gerente irá informar o auditor que há mais um membro a ser auditado. Ambos, gerente e auditor, devem registrar em seu arquivo de *log* a inserção de um novo elemento. Uma nova AD inserida em uma rede de ADs deve registrar em seu arquivo de *log* este evento, prevenindo assim qualquer ação indesejada.

Cada AD deverá possuir um administrador, sendo este responsável pelas configurações de uma AD em particular, como publicação do ponto de confiança, *backup* do banco de dados interno da AD entre outras tarefas. Dependendo da quantidade de ADs, um administrador poderá realizar as tarefas pertinentes a ele em uma ou mais ADs e eventualmente em todas as ADs de uma mesma infra-estrutura.

6.7 Conclusão

A infra-estrutura proposta resolve o problema de organizações que possuem uma grande demanda de protocolação de documentos eletrônicos, visto que não há mais ADs relacionando-se apenas com os clientes, mas sim estão também trocando informações entre si e sendo monitoradas por um gerente. Documentos protocolados em diferentes ADs de uma mesma organização podem ser comparados levando-se em conta apenas o aspecto da datação relativa, o que não pode ser feito com ADs isoladas, ou

melhor, ADs que não fazem a Protocolação Cruzada.

A infra-estrutura aqui explanada define alguns detalhes que não foram tratados em qualquer outro trabalho, como formato do recibo de um documento protocolado em uma infra-estrutura de protocolação digital, a distribuição das entidades, gerência, como duas ADs se relacionam entre outros aspectos.

O problema de comparar temporalmente dois documentos protocolados em ADs diferentes baseando-se apenas no encadeamento foi em parte resolvido, já que nada se pode afirmar de documentos protocolados no mesmo intervalo entre duas protocolações cruzadas. Uma possível deficiência seria a busca por um documento protocolado em uma infra-estrutura que possui muitas ADs e protocolações cruzadas. Dependendo do número de ADs envolvidas, a complexidade desta busca poderia ser tamanha que a tornaria inviável. Para uma comparação nestas condições, pode-se utilizar a datação absoluta, levando-se em conta que um auditor está supervisionando o tempo de todas as ADs envolvidas na infra-estrutura.

Capítulo 7

Considerações Finais

Com o crescente uso de documentos eletrônicos, a demanda por segurança e confiabilidade em transações relacionadas aos mesmos vem aumentando acentuadamente. Como nos documentos tradicionais, um elemento importante para que um documento seja considerado confiável é o carimbo de tempo. Uma Autoridade de Datação (AD) provê este serviço.

Entretanto, uma única AD pode não ser suficiente para suprir as necessidades de uma organização. Para resolver este problema, várias ADs podem ser utilizadas isoladamente. Todavia, esta forma de organização se tornaria difícil de monitorar, visto que as ADs estão trabalhando independentes sem qualquer ligação entre elas ou com um gerente. Outro problema é a comparação temporal realizada em documentos protocolados em diferentes ADs, tanto para a datação relativa quanto para a absoluta. Este trabalho propõe uma infra-estrutura que resolve estes problemas, devido ao fato das ADs estarem ligadas entre si, de haver um gerente gerenciando-as, do relógio das ADs ser auditável e estarem sincronizados com uma fonte de tempo confiável.

Para sincronizar os relógios das ADs existe o Protocolo de Tempo em Rede (*Network Time Protocol* - NTP), o qual permite realizar o sincronismo entre relógios através da Internet com a resolução necessária. Foi verificado, entretanto, que o protocolo NTP não foi projetado para atender aos requisitos de segurança necessários ao serviço de protocolação de documentos eletrônicos.

Assim, são propostas algumas melhorias no protocolo NTP, adicionando maior segurança na troca de mensagens entre as entidades envolvidas no processo de sincronismo. As melhorias propostas não modificam o funcionamento normal do NTP, apenas agregam funcionalidades e segurança ao mesmo. É adicionado ao NTP a autenticação entre servidor e cliente de tempo utilizando certificados emitidos por Autoridades Certificadoras (ACs) externas e também é proposto o auditor de tempo, o qual monitora os relógios das ADs. As melhorias adicionadas ao NTP foram validadas formalmente com métodos formais baseados em máquinas de estado - Redes de Petri - e parcialmente em lógica - lógica GNY.

Na infra-estrutura proposta, além da autenticação e do auditor de tempo, são definidos elementos, protocolos, gerência, organização, configuração inicial e o formato do recibo de um documento protocolado em uma AD pertencente à esta infra-estrutura.

A infra-estrutura é monitorada tanto pelo gerente - responsável pelo estado das ADs - quanto pelo auditor de tempo - responsável pelo tempo utilizado pelas ADs. Qualquer problema em relação à AD e ao tempo, os administradores da infra-estrutura são avisados, além dos relatórios gerados e armazenados. Evidências emitidas pelo gerente e pelo auditor podem ser de grande importância em caso de disputas onde estão envolvidos documentos eletrônicos.

Dois artigos foram publicados, resultados deste trabalho. Um artigo foi publicado em um evento nacional (Simpósio Brasileiro de Redes de Computadores - SBRC 2003) e o outro em um evento internacional (*The 3rd IEEE Latin American Network Operations and Management Symposium* - LANOMS 2003).

7.1 Trabalhos Futuros

Uma questão a ser explorada é o caso do cliente de protocolação não confiar no ponto de confiança que a AD que este utiliza possui. Neste caso, o cliente poderia escolher um ponto de confiança que ele confie e pedir para a sua AD fazer uma protocolação cruzada com a AD que possui o ponto de confiança desejado. Assim, quando

este determinado cliente solicitar protocolação, a AD deverá enviar no recibo do cliente não o encadeamento até seu ponto de confiança mas sim até o ponto de confiança da outra AD, criando assim uma “AD virtual”, visto que fisicamente esta “AD virtual” não existe pois possui parte do encadeamento de duas ADs físicas. Esta proposta pode ser estudada e resolvidos os problemas de busca de documentos e de desempenho agregados a este método, que, em meu ponto de vista, não são tarefas triviais pelo fato de existir uma enorme possibilidade de caminhos a serem analisados.

Um outro trabalho futuro é a validação formal e a implementação da parte que ainda não foi validada e implementada da infra-estrutura proposta neste trabalho. Em se tratando de sincronismo de tempo essa tarefa já foi feita.

Uma infra-estrutura ideal seria a que várias ADs trabalhassem de tal forma que ao enviar um documento a ser protocolado para um determinado endereço, um cliente receberia resposta o mais rápido possível, independente do quão atarefadas então as ADs. As ADs iriam se monitorar de tal forma que quando uma delas estivesse sobrecarregada, uma parte de seu trabalho seria realizada por outra AD e assim por diante. Elas trabalhariam de forma que aparentasse que apenas uma AD estivesse protocolando documentos. Este é um bom tema para ser estudado.

Interessante seria que uma AD descobrisse qual é a sua localização (*time-zone*) para que no sincronismo, o administrador não precisasse configurá-la. A AD poderia ser independente de maneira que quando esta fosse transportada para outra região ou fosse ligada, ela se auto-configurasse e utilizasse o tempo correto segundo sua localização.

Já iniciado em parte nesta dissertação, propõe-se a criação de uma nova RFC parecida com a RFC3161. Esta nova RFC especificaria um novo protocolo de datação para a infra-estrutura de protocolação digital. Na RFC estariam inclusos o método da árvore sincronizada e a protocolação cruzada que foi abordada neste trabalho. O formato da requisição de protocolação e o recibo já foram definidos no capítulo 6, no formato ASN.1. Para a nova RFC, basta apenas refazer o texto da RFC original e criar novas mensagens para o protocolo original se adaptar à infra-estrutura de protocolação digital de documentos eletrônicos.

Referências Bibliográficas

ADAMS, C. et al. *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*. [S.l.], August 2001. Request for Comments: 3161.

AURÉLIO. *Novo Dicionário Aurélio - Século XXI*. [S.l.]: Editora Nova Fronteira, 1999. Dicionário Aurélio Eletrônico Século XXI - Versão 3.0.

BAYER, D.; HABER, S.; STORNETTA, W. S. Improving the efficiency and reliability of digital time-stamping. In: . [s.n.], 1992. p. 329–334. Disponível em: <citeseer.nj.nec.com/bayer93improving.html>.

BENALOH, J.; MARE, M. de. *Efficient Broadcast time-stamping*. 1991. Disponível em: <citeseer.nj.nec.com/benaloh91efficient.html>.

BENALOH, J. C.; MARE, M. de. One-way accumulators: A decentralized alternative to digital signatures. *Lecture Notes in Computer Science*, v. 765, p. 274–??, 1994. Disponível em: <citeseer.nj.nec.com/304083.html>.

BORTOLI, D. L. *O Documento Eletrônico No Ofício de Registro Civil de Pessoas Naturais*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina - Programa de Pós Graduação em Ciências da Computação, Florianópolis - SC, 2002.

BRASIL. *A Política de Governo Eletrônico No Brasil*. [S.l.], Agosto 2001. Disponível em: <http://www.iberomunicipios.org/docs/brasil.pdf>.

BRASIL. *Decreto Nº 4.264, de 10 de Junho de 2002*. Junho 2002. Restabelece O Regulamento Aprovado Pelo Decreto Nº 10.546 de 5 de Novembro de 1913.

BULDAS, A.; LAUD, P. New linking schemes for digital time-stamping. In: *Information Security and Cryptology*. [s.n.], 1998. p. 3–13. Disponível em: <citeseer.nj.nec.com/article/buldas98new.html>.

BULDAS, A. et al. Time-stamping with Binary Linking Schemes. In: KRAWCZYK, H. (Ed.). *Advances on Cryptology - CRYPTO '98*. Santa Barbara, USA: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v. 1462), p. 486–501. Disponível em: <citeseer.nj.nec.com/buldas98timestamping.html>.

BULDAS, A.; LIPMAA, H.; SCHOENMAKERS, B. Optimally efficient accountable time-stamping. In: *Public Key Cryptography - pkc 2000*. Melbourne, Australia: [s.n.], 2000. Vol. 1751 of Lecture Notes in Computer Science, p. 293–305. Disponível em: <citeseer.nj.nec.com/buldas00optimally.html>.

BURROWS, M.; ABADI, M.; NEEDHAM, R. M. A logic of authentication. *ACM Transactions on Computer Systems*, v. 8, n. 1, p. 18–36, February 1990. A Formal Semantics for Evaluating Cryptographic Protocols p 14.

CALLAS, J. et al. *OpenPGP Message Format*. [S.l.], November 1998. Request for Comments: 2440.

CASE, J. et al. *A Simple Network Management Protocol (SNMP)*. [S.l.], May 1990. Request for Comments: 1157.

COSTA, V. *Um estudo da Confiabilidade do processo de protocolação digital de documentos eletrônicos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2003.

DEETHS, D.; BRUNETTE, G. *Using NTP to Control and Synchronize System Clocks - Part I: Introduction to NTP*. July 2001. Sun BluePrints OnLine.

DEETHS, D.; BRUNETTE, G. *Using NTP to Control and Synchronize System Clocks - Part III: NTP Monitoring and Troubleshooting*. September 2001. Sun BluePrints OnLine.

DIAS, J.; CUSTÓDIO, R. F.; DEMÉTRIO, D. B. Sincronização segura de relógio para documentos eletrônicos. *Simpósio Brasileiro de Rede de Computadores 2003*, Maio 2003.

DIAS, J. et al. Reliable clock synchronization for electronic documents. *The 3rd IEEE Latin American Network Operations and Management Symposium (LANOMS'2003)*, September 2003. Iguassu Falls, Brazil.

GONG, L.; NEEDHAM, R.; YAHALOM, R. Reasoning about belief in cryptographic protocols. *Proceedings of the IEEE 1990 Symposium on Security and Privacy*, p. 234–248, February 1990. University of Cambrige Computer Laboratory, England.

HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. *Lecture Notes in Computer Science*, v. 537, p. 437–??, 1991. Disponível em: <citeseer.nj.nec.com/haber91how.html>.

I'ANSON, C.; MITCHELL, C. Security defects in CCITT recommnedation X.509 - the directory authentication framework. *Computer Communications Review*, April 1990.

IGNACZAK, L. *Um Novo Modelo de Infra-estrutura de Chaves Públicas para Uso no Brasil Utilizando Aplicativos com o Código Fonte Aberto*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2002.

IRIG. *IRIG Standard 205-87*. 1987. Range Commanders Council of the US Army White Sands Missile Range.

ITUT. *Specification of Abstract Syntax Notation One (ASN.1)*. [S.l.], 1988. Recommendation X.208.

ITUT. *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*. [S.l.], 1988. Recommendation X.209.

JUST, M. K. On the temporal authentication of digital data. Carleton University, December 1998.

LABSEC. *Laboratório de segurança em computação*. Março 2003. Universidade Federal de Santa Catarina, Brasil. Disponível em: <http://www.labsec.ufsc.br>.

LIPMAA, H. *Digital Time-Stamping*. March 2003. Cryptology Pointers by Helger Lipmaa. Disponível em: <http://saturn.tcs.hut.fi/ helger/crypto/link/timestamping/>.

LOMBARDI, A.; LIPMAA, H. Digital signatures, timestamping and the corresponding infrastructure. January 1998.

LOMBARDI, M. *Computer Time Synchronization*. [S.l.], 2002.

MARCACINI, A. T. R. *O Documento Eletrônico como meio de Prova*. [S.l.], Março 1999.

MATHURIA, A.; SAFAVI-NAINI, R.; NICKOLAS, P. Some remarks on the logic of Gong, Needham and Yahalom. *Proceedings of the International Computer Symposium*, v. 1, p. 303–308, December 1994. Departament of Computer Science, University of Wollongong.

MAURER, U. M.; WOLF, S. The Diffie-Hellman protocol. *Designs, Codes and Cryptography*, v. 19, p. 147–171, 2000. Disponível em: <citeseer.nj.nec.com/maurer99diffiehellman.html>.

MEADOWS, C. A. Formal verification of cryptographic protocols: A survey. 1994. Center for High Assurance Computer Systems, Naval Research Laboratory, EUA.

MENEZES, A. J.; OORSCHOT, P. C. van; VANSTONE, S. A. *Handbook of Applied Cryptography*. [S.l.]: CRC Press, 1996.

MILLS, D. L. *Network Time Protocol (Version 3) - Specification, Implementation and Analysis*. March 1992. Request for Comments: 1305. Disponível em: <http://rfc.net/rfc1305.html>.

MILLS, D. L. *Simple Network Time Protocol (SNTP)*. March 1995. Internet draft of Internet Engineering Task Force (IETF) - Request for Comments 1769. Disponível em: <http://www.faqs.org/rfcs/rfc1769.html>.

MILLS, D. L. Cryptography authentication for real-time network protocols. *AMS DIMACS - Series in Discrete Mathematics and Theorical Computes Science*, v. 45, p. 135–144, 1999.

MILLS, D. L. *The Autokey Security Architecture,Protocol and Algorithms*. [S.l.], Fevereiro 2003. Technical Report.

MILLS, D. L. *NTP Algorithm Analysis*. Junho 2003. Apresentation Notes. Disponível em: <www.eecis.udel.edu/ mills/database/ brief/algor/algor.ppt>.

MILLS, D. L. *NTP Architecture, Protocol and Algorithms*. Junho 2003. Apresentation Notes. Disponível em: <www.eecis.udel.edu/ mills/database/brief/arch/arch.ppt>.

MILLS, D. L. *NTP Security Model*. Junho 2003. Apresentation Notes. Disponível em: <http://www.eecis.udel.edu/ mills/database/brief/keys/keys.ppt>.

MILLS, D. L. *NTP: The Network Time Protocol*. April 2003. University of Delaware. Disponível em: <http://www.ntp.org>.

MITCHELL, C.; WALKER, M.; RUSH, D. CCITT/ISO Standards for Secure Message Handling. *IEEE Journal on Selected Areas in Communications*, May 1989.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 1989.

NIST. *Time and Frequency Division - Services*. May 2003. National Institute of Standards and Technology. Disponível em: <http://www.boulder.nist.gov/timefreq/>.

NOTOYA, A. E. *IARSDE - Infra-Estrutura de Armazenamento e Recuperação Segura de Documentos Eletrônicos: Validade de documento eletrônico por tempo indeterminado*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2002.

ON. *Observatório Nacional - Divisão Do Serviço da Hora (DSH)*. Fevereiro 2003. Disponível em: <http://pcdsh01.on.br>.

PACKARD, H. *The Science of Timekeeping*. [S.l.], March 2003. Application Note 1289. Disponível em: <http://www.allanstime.com/Publications/DWA/Science_Timekeeping/TheScienceOfTimekeeping.pdf>.

PALKO, R. H. *Securing Time - The Autokey Protocols*. August 2001. Disponível em: <http://www.sans.org/rr/protocols/autokey.php>.

PASQUAL, E. S. *IDDE - Uma Infra-Estrutura Para a Datação de Documentos Eletrônicos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2002.

PASQUAL, E. S.; DIAS, J. D. S.; CUSTÓDIO, R. F. Árvore sincronizada - um método para datação de documentos eletrônicos. 2002.

PASQUAL, E. S.; DIAS, J. D. S.; CUSTÓDIO, R. F. A new method for digital time-stamping of electronic document. 2002. Hawaii, EUA.

POSTEL, J. *Daytime Protocol*. May 1983. Internet draft of Internet Engineering Task Force (IETF) - Request for Comments 867. Disponível em: <http://www.faqs.org/rfcs/rfc867.html>.

POSTEL, J.; HARRENSTIEN, K. *Time Protocol*. May 1983. Internet draft of Internet Engineering Task Force (IETF) - Request for Comments 868. Disponível em: <http://www.faqs.org/rfcs/rfc868.html>.

ROOS, M. *Integrating Time-Stamping and Notarization*. Dissertação (Mestrado) — Tartu Data Security Lab, Küberneetika AS, June 1999.

SCHAEFER, E. A simplified Data Encryption Standard algorithm. *Criptologia*, January 1996.

STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. 2 ed. [S.l.]: Prentice-Hall, Inc, 1998.

STINSON, D. R. *Cryptography: Theory and Practice*. [S.l.]: CRC Press, 1995.

VEIGA, L. A. O. D. *Direito e Internet: Aspectos Jurídicos Do Documento Eletrônico*. [S.l.], 2002.

Apêndice A

Glossário

Assinatura Digital: Permite a autenticação de mensagens digitais, assegurando ao destinatário de qualquer mensagem digital a identidade de quem enviou e a integridade da mensagem (BORTOLI, 2002).

Autoridade Certificadora (AC): Autoridade responsável por emitir e assinar certificados digitais. Pode também revogar certificados digitais.

Autoridade de Datação (AD): Autoridade responsável por datar documentos eletrônicos utilizando métodos absolutos e/ou relativos de datação.

Base Gerencial de Informações de Protocolação (BGIP): Base de dados onde ficam armazenadas as variáveis que representam o estado dos objetos gerenciáveis de uma Autoridade de Datação.

Carimbo de tempo: É o registro de tempo anexado ao documento eletrônico emitido por uma Autoridade de Datação.

Certificado Digital: É um documento eletrônico assinado por uma Autoridade Certificadora (AC) que contém o nome do proprietário, a entidade emissora, a chave pública do proprietário e o período de validade do certificado.

Criptografia Assimétrica: Tipo de criptografia que utiliza duas chaves diferentes para cifrar e decifrar mensagens. O que uma cifra a outra decifra e vice-versa.

Criptografia Simétrica: Utiliza uma senha ou um par de chaves idênticas para cifrar e decifrar mensagens.

Data: Um instante específico definido em uma escala de tempo determinada. **NOTA**: A data pode ser convencionalmente expressa em anos, meses, dias, horas, minutos, segundos e frações. Data Juliana (JD) e Data Juliana Modificada (MJD), são também medidas úteis para definição de datas (ON, 2003).

Datação: Criação de uma âncora temporal em um documento. Em documentos eletrônicos pode ser feita de duas maneiras: anexando-se o tempo ao documento eletrônico ou encadeando um documento ao outro para que a seqüência entre eles seja mantida.

Datação Absoluta: Criação de uma âncora temporal anexando-se o tempo absoluto ao documento eletrônico.

Datação Relativa: Criação de uma âncora temporal em um documento eletrônico encadeando um documento ao outro para que a seqüência de sua chegada à AD seja mantida

DES: O algoritmo *Data Encryption Standard* é utilizado para cifrar mensagens. Utiliza uma chave de 56 bits e os dados são cifrados em blocos de 64 bits (SCHAEFER, 1996).

Diffie-Hellman: Algoritmo de troca de chaves segura. Seu objetivo é habilitar dois usuários trocar suas chaves seguramente para serem utilizadas em algum algoritmo de cifragem de mensagens (MAURER; WOLF, 2000).

Encadeamento binário: Esquema binário de encadeamento, que pode ser definido como sendo um grafo direcionado não cíclico, onde todos os vértices tem pelo menos duas arestas (PASQUAL, 2002).

Exatidão: O grau de conformidade de um valor medido ou calculado em relação à sua definição ou com respeito a uma referência padrão (ON, 2003).

Função Resumo (*Hash*): Algoritmo que tem como entrada qualquer tamanho de bloco e como saída um bloco de tamanho fixo. É inviável obter a mensagem original através do resumo.

Hora Média de Greenwich (GMT): (*Greenwich Mean Time*) Um sistema de 24 horas baseado na hora Solar média mais 12 horas em Greenwich, Inglaterra. A Hora Média de Greenwich pode ser considerada aproximadamente equivalente ao Tempo Universal Coordenado (UTC), o qual é disseminado por todas rádio emissoras de tempo e freqüência. Entretanto, GMT é um termo obsoleto e foi substituído por UTC (ON, 2003).

Infra-Estrutura de Chaves Públicas (ICP): A ICP consiste em uma rede de protocolos, padrões e serviços, para suportar aplicações de criptografia de chaves públicas (IGNACZAK, 2002).

MD5: O *Message Digest Algorithm* é uma função resumo que possui como saída uma string de 128 bits.

Método da árvore: Forma relativa de datação de documentos eletrônicos baseada em um grafo e seu principal objetivo é criar unidades de tempo denominadas rodadas.

Método da árvore sincronizada: Forma relativa de datação de documentos eletrônicos baseado no método da árvore com a adição do conceito de saltos, que englobam várias rodadas.

Método do encadeamento linear: Forma relativa de datação de documentos eletrônicos baseado em uma lista.

OpenSSL: Ferramenta de código fonte aberto que implementa os protocolos de Camada de Conexão Segura (*Secure Sockets Layer* - SSL) e de Segurança na Camada de Transporte (*Transport Layer Security* - TLS), bem como uma biblioteca de criptografia.

Precisão: O grau de concordância mútua entre uma série de medidas individuais. A precisão é muitas vezes, mas não necessariamente, expressa pelo desvio padrão das medidas (ON, 2003).

Protocolação: É a utilização de formas absolutas e relativas de datação (datação híbrida). O recibo de protocolação contém o tempo absoluto, a seqüência de encadeamento do documento em questão até o próximo ponto de confiança e contém a assinatura da Autoridade de Datação.

Protocoladora Digital de Documentos Eletrônicos (PDDE): Nome denominado a uma AD por Pasqual (2002) quando esta utiliza especificamente o Método da Árvore Sincronizada como forma de datação relativa.

Protocolo Simples de Gerenciamento de Protocolação Digital (PSGPD): Protocolo de gerenciamento de uma infra-estrutura de protocolação digital. É através deste protocolo que gerente e agentes se comunicam.

Recibo de datação: Ou recibo de protocolação é um documento que garante que um documento foi datado em uma certa hora e data por uma Autoridade de Datação. Para a datação relativa, o recibo contém informações para que o encadeamento possa ser recalculado. Na datação híbrida, além das informações do encadeamento, estão disponíveis informações sobre o tempo absoluto.

RSA: O algoritmo *Rivest-Shamir-Adleman* utiliza a criptografia assimétrica para cifrar e decifrar mensagens.

Segundo intercalado: (*Leap Second*) Uma mudança de tempo intencional de um segundo, usado para ajustar o UTC para assegurar uma concordância aproximada com o UT1. A inserção de um segundo é chamada de segundo intercalado positivo, e a omissão de um segundo é chamada de segundo intercalado negativo. Um segundo intercalado positivo tem sido necessário aproximadamente uma vez por ano (ON, 2003).

Sincronização: O processo de medida da diferença em tempo entre duas escalas de tempo, tal como os sinais de saída gerados por dois relógios. No contexto de *timing*, entende-se como sincronização colocar em fase dois relógios ou fluxo de dados tal que sua diferença seja zero (ON, 2003).

Tempo Universal Coordenado (UTC): (*Universal Time Coordinated*) Uma escala de tempo coordenada, mantida pelo *Bureau* Internacional de Pesos e Medidas (BIPM), que constitui a base de uma disseminação coordenada de freqüências padrão e sinais horários.

X.509: Tipo de estrutura de certificados digitais especificado pela ITUT. É baseado em criptografia assimétrica e assinatura digital (I'ANSON; MITCHELL, 1990; MITCHELL; WALKER; RUSH, 1989).

Apêndice B

Implementação

A seguir é explicado com detalhes como foi feita a implementação da autenticação entre servidor e cliente de tempo e a implementação do programa auditor.

A implementação foi realizada pelos alunos de graduação em Ciência da Computação da UFSC: Vitor Claudino dos Santos e Bruno Leonardo Martins de Melo.

A implementação foi feita utilizando-se a ferramenta PHP e o banco de dados MySQL.

B.1 Autenticação entre cliente e servidor de tempo

Na versão 4 do NTP (*Network Time Protocol*), a autenticação entre os participantes do esquema de sincronismo é feita através do protocolo *Autokey*, que é detalhadamente explicado no capítulo 4. Este protocolo tem o objetivo de fazer com que o servidor de tempo seja conhecido pelos clientes, através de certificados digitais auto-assinados gerados pelo próprio NTP. Esse modelo possui duas limitações: o fato de não haver autenticação do cliente perante o servidor e o fato de não ser possível utilizar certificados digitais emitidos por uma AC externa. Visando contornar esses problemas, foi implementado um modelo para a autenticação das entidades envolvidas no processo de sincronismo, como pode ser visto a seguir.

B.1.1 Adaptação ao *Autokey*

A implementação é baseada na evolução do código NTP e na capacidade dos clientes fornecerem seus tempos apenas para fins de consulta. Assim, para que clientes e servidores fossem autenticados um perante ao outro, considera-se que os clientes são servidores de tempo, temporariamente, para que estes possam se autenticar perante o servidor. Entretanto, o servidor de tempo possui um arquivo de configuração contendo os endereços dos clientes que estão se fazendo passar por servidor, isto para que o tempo destes clientes não seja tomado como base para o servidor de tempo. O auditor tem sempre uma fonte de tempo confiável, na qual este irá se basear para fazer a auditoria. Na solicitação de tempo feita pelo auditor, este irá primeiramente verificar se o *stratum* da entidade auditada é maior do que o *stratum* do auditor. Caso essa possibilidade se torne verdadeira, a auditoria não acontece.

Considerando as alterações descritas acima, o protocolo *Autokey* resolve o problema da autenticação entre cliente e servidor. Para que certificados emitidos por ACs externas possam ser utilizados, estes devem seguir o padrão:

- O certificado deve seguir a sintaxe ASN.1;
- Deve ser do tipo X.509, versão 3, codificado no formato PEM;
- O tamanho do certificado codificado em ASN.1 não deve ultrapassar 1024 bytes;
- O campo "*subject_cn*" do certificado deve conter o nome qualificado do *host* no qual o certificado está sendo utilizado;
- Outros campos "*subject*" são ignorados;
- Os campos de extensão do certificado não devem conter um campo "*subject key identifier*" ou "*issuer key identifier*", entretanto o campo "*extended key usage*" deve conter o valor *trustRoot* para especificar um *host* confiável;
- Outros campos de extensão são ignorados.

As operações de autenticação no NTP podem utilizar tanto chave simétrica quanto assimétrica. O foco desse trabalho está no uso da criptografia assimétrica para operações de autenticação, sendo que no NTP o protocolo *Autokey* é o responsável pela autenticação utilizando chaves assimétricas. O funcionamento do protocolo *Autokey* é baseado na geração dos certificados digitais por um utilitário do NTP, o *ntp-keygen* e a distribuição das chaves é efetuada seguindo o protocolo Diffie-Hellman. A verificação dos certificados é realizada através de chamadas à biblioteca RSAREFF e a verificação da integridade das mensagens é garantida com o uso da função resumo MD5.

A Figura B.1 mostra como acontece o cadastro de entidades a serem auditadas e se estas entidades deverão ser autenticadas com certificado digital ou não.

Figura B.1: Cadastro de entidades a serem auditadas.

B.2 Auditoria

O auditor é um programa hospedado em um servidor de tempo e tem como principal função monitorar os relógios das entidades envolvidas no processo de sincronismo, fazendo com que entidades que possuam relógios defeituosos ou que atuem de forma maliciosa tenham seu tempo corrigido ou em casos extremos, sejam excluídos da rede NTP. O programa auditor tem o poder de solicitar a revogação do certificado de qualquer equipamento que apresente erros acima dos padrões permitidos. Esses padrões serão discutidos mais adiante.

A identidade do programa auditor fica camuflada pela entidade que o hospeda. Para tanto, a entidade hospedeira deve estar devidamente autenticada através de um certificado digital. Assim, quando uma entidade é auditada, esta não poderá saber se é o auditor que está requisitando o tempo para auditoria ou uma entidade qualquer desejando fazer sincronismo.

O processo de análise dos tempos baseia-se no fato de que a entidade hospedeira esteja com seu relógio sincronizado ao UTC (*Universal Time Coordinated* - Tempo Universal Coordenado), ou seja, que este servidor esteja conectado a uma fonte de tempo confiável. A Figura B.2 mostra o cadastro de servidores de tempo nos quais o auditor irá se basear para realizar a auditoria.

Para a auditoria, foi implementado um módulo que controla a configuração e monitora o *ntpd*[1]. O *ntpd* se encarrega de fazer requisições de tempo às entidades sob auditoria, em um intervalo de tempo fixo, e armazena as respostas localmente. Assim, quando o auditor deseja resgatar o tempo de determinada entidade este acessa sua memória através dos comandos *NTPq* e *NTPdc*, pertencentes ao NTP. Toda parte de requisição de tempo fica a cargo do *ntpd*, que é configurado com base nos equipamentos cadastrados no sistema de administração do auditor, como mostrado na Figura B.1.

[1] *ntpd*: é o *deamon* NTP que roda em segundo plano e é encarregado de todo processo de sincronismo e autenticação, utilizando para isso complexos algoritmos para seleção de servidores e métodos de criptografia por chave simétrica e assimétrica. Toda configuração acerca do funcionamento do *ntpd* como servidores, chaves para autenticação, controle de acesso, entre outras configurações, é especificada no arquivo de configuração do *ntpd*.

Figura B.2: Cadastro dos servidores de tempo do auditor.

A implementação do auditor é dividida em duas partes:

- **Sistema de Administração**: Interface web de onde é possível realizar todas as operações de controle, manutenção e cadastro necessários para a auditoria;
- **Programa Auditor**: Programa rodando em *background*, encarregado de realizar as requisições de tempo consultando os resultados de requisições já realizadas pelo *ntpd*. Como já citado, o auditor acessa os tempos através do comando *ntpq*.

De forma mais prática, o auditor é um programa que envia requisições de tempo periodicamente às entidades sob auditoria. Os resultados das requisições do auditor contêm o tempo da entidade sob auditoria. Ao mesmo tempo em que a entidade retorna sua hora local, o auditor verifica sua própria hora interna e realiza uma comparação

entre os tempos e em caso de uma diferença fora dos padrões especificados, o número de erros desta determinada entidade vai aumentando até um limite máximo pré-configurado, fazendo com que ela não seja mais considerada apta a fornecer tempo a seus clientes e seu certificado digital é revogado pela AC.

O administrador do auditor deve configurar a quantidade máxima de erros que o relógio de uma entidade sob auditoria pode possuir em relação ao tempo da fonte. Deve configurar também a tolerância máxima da diferença entre o tempo coletado e o tempo da fonte, o intervalo de tempo que o auditor irá requisitar o tempo para as entidades auditadas e o e-mail do administrador deve ser informado para que este seja avisado em caso de irregularidades. A Figura B.3 mostra como estas configurações são feitas.

Figura B.3: Configurações dos parâmetros de auditoria.

O auditor fica rodando constantemente, até que o administrador do mesmo resolva reiniciá-lo. A interface de gerência é mostrada na Figura B.4.

Figura B.4: Gerência do auditor e do *ntpd*.

Para tornar possível a análise posterior de cada entidade auditada, todas as requisições feitas a cada entidade são adicionadas a um arquivo de *log* que deve ser protocolado em determinado intervalo de tempo, garantindo que este *log* não seja modificado. Atualmente o processo de protocolação dos *logs* está apenas indicado no código fonte do auditor, devido a alguns problemas encontrados na formatação das requisições e dos recibos. Também ainda não foi implementado a emissão de alvará às entidades auditadas pelo auditor. A Figura B.5 mostra um exemplo de *log* do auditor.

Pode-se também consultar o status de todas as entidades sob auditoria ou de uma entidade específica.

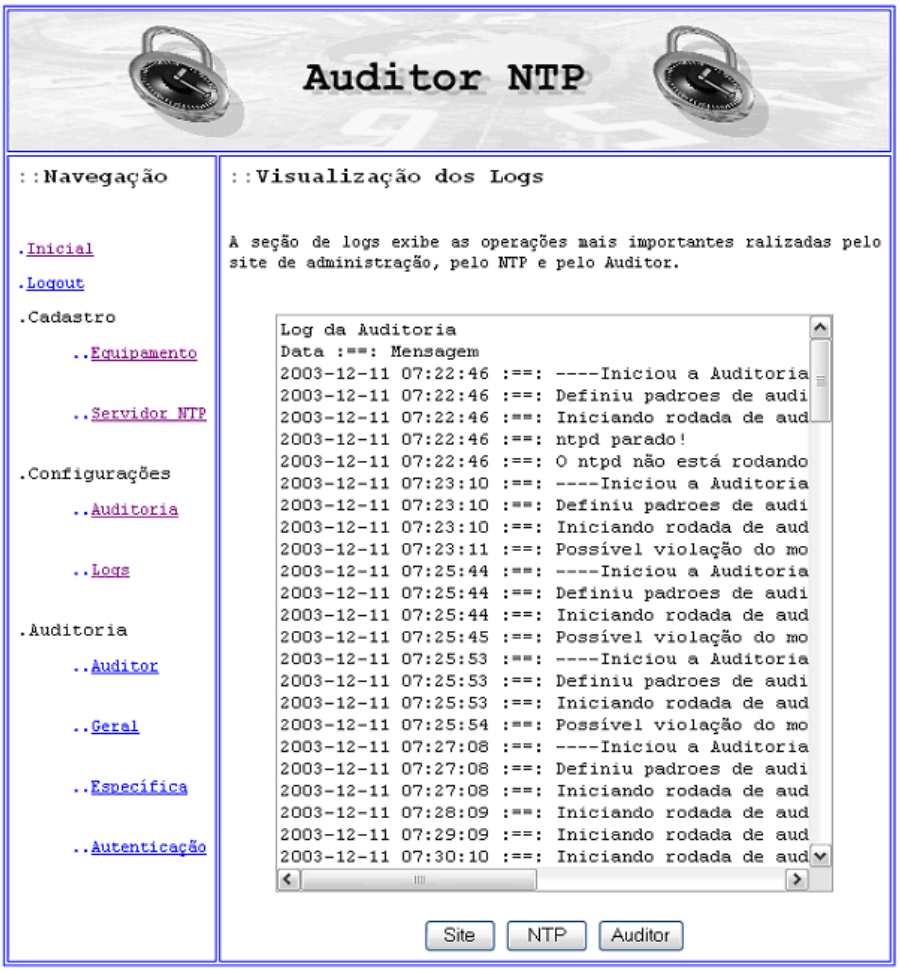

Figura B.5: Exemplo de *log* do auditor.

Apêndice C

Publicações

No decorrer do período de mestrado, foram publicados dois artigos relativos ao tema de estudo da dissertação.

O primeiro artigo publicado, intitulado como “Sincronização Segura de Relógio para Documentos Eletrônicos”, foi publicado em maio de 2003 no 21^{o} Simpósio Brasileiro de Rede de Computadores 2003 (SBRC’2003), tendo como autores: Julio da Silva Dias, Ricardo Felipe Custódio e Denise Bendo Demétrio.

O segundo artigo, intitulado como “*Reliable Clock Synchronization for Electronic Documents*”, foi publicado em setembro de 2003 no *3rd IEEE Latin American Network Operations and Management Symposium* (LANOMS’2003), tendo como autores: Julio da Silva Dias, Denise Bendo Demétrio, Ricardo Felipe Custódio e Carlos Roberto de Rolt.

A seguir são apresentados os dois artigos mencionados.